Formal Analysis of State Estimation for Nonlinear Model Predictive Control

By:
Sekhonyana Moeti

A dissertation submitted to the Department of Electrical Engineering,
University of Cape Town, in fulfillment of the requirements
for the degree of Master of Science in Engineering.

Supervised by:
Mr. Tsoeu Mohohlo
Dept. of Electrical Engineering
University of Cape Town

November 27, 2015

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the degree of Master of Science in Engineering in the University of Cape Town. It has not been submitted before for any degree or examination in any other university.

Signature of Author .....

Cape Town

15 February 2015

Abstract

The main goal of this study is to carry out a closed-loop performance analysis of state estimation methods when implemented in the formulation of nonlinear model predictive control. The analysis is facilitated by two nonlinear optimal state estimation methods: augmented state EKF (ASEKF) and augmented state UKF (ASUKF) for comparison purposes. Each state estimation method is coupled to the same NMPC controller to form state estimation-based NMPC controllers, that is, to form the ASEKF-NMPC and ASUKF-NMPC controllers. The resulting NMPC controllers are applied for position control of the magnetic levitation system to validate their closed-loop performances. The ASEKF-NMPC and ASUKF-NMPC controllers are further applied for the angular position control of the inverted pendulum mounted on a cart system for comparative analysis. The controlled system is perturbed with different error sources: output step disturbance and 5% parametric plant-model mismatch. Output step disturbance is introduced to the system to disturb the pendulum from its upright position while the 5% mismatch is applied to the parameters of the model of the controlled system throughout the simulation.

To facilitate fair analysis, Pareto front ranking method is chosen as an evaluation method whereby the cost functions are defined according to the author's preferences. The cost functions served as performance markers for analyzing performance of ASEKF and ASUKF in NMPC formulation in multidimensional space. The tunable parameters of the ASEKF-NMPC and ASUKF-NMPC controllers are chosen to be the decision variables of the evaluation problem. The state estimation methods are evaluated in terms of estimation accuracy, system's response time, peak overshoot and control performance. The *Level Diagrams* tool is used for good visualization of the Pareto fronts to evaluate which estimator performs better in the closed-loop. Finally, the points on the *Level Diagrams* which provide a performance closest to the desired are selected and tested through simulation runs on the inverted pendulum on a moving cart system.

Acknowledgments

This thesis would not have been possible without an unwearying support and help of many people. First, I would like to express sincere gratitude to my supervisor, Mr. Tsoeu Mohohlo, for his patient guidance, invaluable comments and advices. He did not only help me to make this work a success, but he also inspired me greatly to develop much interest in the field of control engineering. I feel so lucky to have worked with him as a supervisor. I would also like to thank my colleagues for the motivation they gave me throughout the duration of this work. Finally, I wish to avail myself of the opportunity to thank my family and friends for their support and endless love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter is intended to provide a brief background of what this thesis is about. Most importantly, it provides a statement of a problem to be addressed here. Furthermore, a purpose of study, objectives, scope and limitations are discussed. The chapter is closed by providing a thesis outline which briefs on what is been covered in the following chapters.

1.1 Background

Traditionally, a control problem is formulated such that a current state of a controlled system tracks a desired state trajectory as best as possible, in the presence of unknown disturbances and desired state trajectory changes. That is, if the current state is far away from a reference trajectory, appropriate control action is taken to steer the state towards the reference trajectory. If the current state is already close enough to the reference trajectory, appropriate control action is taken to regulate the state there [1], in the presence of disturbances. This control problem is not too complicated to solve. Hence, it has motivated a wide application of classical control schemes such as Proportional-Integral-Derivative (PID) controllers and the clever arrangements of such schemes in process industries [2].

A better understanding of industrial processes has led to a consideration of system constraints and multivariable interactions handling as additional control objectives [2]. This control problem can be complicated to solve. Hence, it can be cumbersome if not impracticable for PID schemes to handle it, especially when a control problem is subjected to stringent constraints [2]. Over the past two decades, Model Predictive Control (MPC) has been proposed as an effective advanced control strategy in process industries. An advantage that it has over PID is its ability to handle system constraints and multivariable interactions naturally within its framework [3, 4, 5]. MPC technology was originally developed to meet specialized control needs for power plants and petroleum refinery applications [6]. Nowadays, it can be found in a wide variety of manufacturing environments including chemicals, food processing, automotive, aerospace, metallurgy, and pulp and paper [6, 7].

MPC works by solving a given optimal control problem (OCP) in real-time to determine the open-loop control variable trajectories and state estimates given process measurements

[8, 9]. Linear MPC (LMPC) schemes have a long history of being among the most effective advanced control schemes in process industries. Despite their long history of success, LMPC schemes are not considered the best candidates for feedback control of systems that are highly nonlinear. To control such systems adequately without changing a basic MPC principle, researchers and designers in academia and industrial communities derived a direct extension of LMPC called Nonlinear Model Predictive Control (NMPC). A formulation of NMPC problem is similar to that of LMPC problem and owing to that, NMPC has inherited most of the essential properties of LMPC such as handling of system constraints and multivariable interactions [5]. However, instead of employing linear system models, NMPC employs nonlinear models for prediction purposes [10]. Linear models do not adequately describe a controlled system except near a point at which the model was identified [9]. On the other hand, nonlinear models capture the behavior of the controlled system accurately and this makes them appealing for control of systems with strong nonlinearities.

One of the major obstacles to implementing NMPC schemes is long computational time that it takes an algorithm not only to make predictions [10] but also to find a solution to a given OCP at each time step. This drawback makes NMPC less appealing for control of systems with fast dynamics. Another obstacle which is also worth mentioning is that, OCPs that are solved in the NMPC formulation maybe nonconvex. Nonconvex problems have multiple locally optimal solutions hence, it may take a long time to identify whether the problem has no feasible solution or an obtained solution is globally optimal. To handle the undesired locally optimal solutions, the problem may be formulated as a constrained OCP and then solved using Sequential Quadratic Programming (SQP) solution methods. Diehln, Ferreau and Haverbeke in [11] review numerical methods for solutions of the OCPs which arise in the NMPC formulations. They focus specifically on SQP and Interior-Point (IP) solution methods. Detailed coverage of NMPC solutions can be found in [1, 12].

To implement MPC algorithms, it is required that an entire state vector of the controlled system be accessible for measurement in real-time [13, 14]. Realistically, only a subset of the system states are observable and all unobservable state variables need to be estimated from noisy system measurements. In addition to that, dynamic models that are utilized for prediction purposes in MPC algorithm can exist with significant plant-model mismatches and unknown disturbances [14]. Therefore, robustness has to be one of the primary objectives when designing NMPC schemes. While several methods have been proposed to handle disturbances and plant-model mismatches, only a few enjoy practical deployment. A most common method is an *augmentation* method, where a disturbance model is appended to the system model and the disturbance states are estimated [5, 12, 14] as well as the system states. The aforementioned problems simply render state estimation problem alongside disturbance estimation, prediction and compensation problems as most important problems to be considered when developing MPC algorithms.

State estimation methods of interest in this thesis are an Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) because they are the competing methods for nonlinear estimation problem according to literature [13, 15, 16]. EKF is a most commonly employed algorithm relative to UKF. It works by propagating the probability distribution

function (pdf) of state estimates analytically through a *linear approximation* of the system model around an operating point in real-time [13]. Unfortunately, linear approximation introduces cumulative estimation errors which may lead to divergence of the algorithm when the system is highly nonlinear. UKF is a derivative-free alternative method. Even though it has seen a few appearances in different applications compared to EKF, it is relatively suitable for highly nonlinear systems. On the contrary, it employs an *unscented transformation* (UT) technique as opposed to linear approximation [17]. Literature suggests that UKF provides improved results compared with EKF, but at a cost of large computational time . Detailed coverage of optimal nonlinear state estimation methods can be found in [18, 15, 19].

Many publications that discuss that implementation of EKF and UKF in NMPC formulation are available already [5, 10, 12, 14, 20, 21]. However, a criteria for choosing either EKF or UKF for a particular controlled system is not obvious. Moreno in [20] conducted a closed-loop performance evaluation of various nonlinear state estimation methods (EKF and UKF included) when integrated into NMPC formulation. He carried out the evaluation with an estimation accuracy (which was quantified by Normalized Root Mean Squared Error (NRMSE)), computational time (which was quantified by CPU time) and control performance (which was quantified by tracking error NRMSE) set as the objectives. A similar task is conducted in this thesis.

1.2 Research Contributions

There are many state estimation methods in literature which can be proposed for designing a state estimation based NMPC for a particular system, but a criteria for choosing the best method is not obvious. A reason is that these methods are formulated differently. For instance, EKF is formulated such that it employs linear approximation technique to propagate the pdf of the states while UKF is employs UT to propagate the pdf of the states. Analyzing these methods from their properties, one may conclude that UKF is more accurate than EKF since it uses UT. On the other hand, one may conclude that EKF outperforms UKF in term of speed of convergence since it uses linear approximation. This may not always be the case for all dynamic systems. Therefore, there is a need to carry out an evaluation which gives a more useful insight on each method through defining appropriate design objectives as performance measures in order to facilitate a formal evaluation.

In this thesis, a closed-loop performance evaluation problem is formulated as a Multiobjective Optimization (MOO) problem. The state estimation methods are evaluated based on the chosen design objectives and from the evaluation results, strengths and weaknesses of each method are *carefully* investigated. The chosen design objectives are a good estimation accuracy, good tracking and disturbance rejection, fast system's response, small peak overshoots and small magnitudes of control inputs.

Evaluating the state estimation methods to be implemented in a design of state esti-

mation based NMPC controller for a given system is important in that it can guide a decision-maker to make informed choice of the method which is more suitable for design. Moreover, evaluation can also help tune the parameters of a chosen method in order to achieve optimal closed-loop performance of the NMPC.

1.3 Problem Statement

To successfully implement NMPC, one of the requirements is that all state variables should be measurable at each sampling instant. For many practical systems this is not always the case. This necessitates implementation of state estimation methods in NMPC formulation to estimate unmeasured state variables. An issue of accuracy of estimation has an impact on the closed-loop performance of the control system. Not only do the states have to be estimated but also the modeling errors: unknown disturbances and plant-model mismatches, which are introduced into the system. This information is require when formulating the OCP, which is then solved to determine an open-loop sequence of control inputs that will drive the system in future time. This situation renders implementing state estimation in NMPC formulation important in order to improve its robustness.

1.4 Purpose of Study

A purpose of this study is to carry out the closed-loop performance evaluation of an augmented state EKF (ASEKF) and augmented state UKF (ASUKF), when implemented for the design of state estimation-based NMPC controller given estimation accuracy, tracking and disturbance rejection, system's response time, maximum peak overshoot and magnitude of control inputs set as design objectives. This work serves as a contribution to an ongoing study on enhancing the closed-loop performance and robustness of various NMPC schemes.

1.5 Objectives of Study

The objectives of this study are

- to develop and validate augmented state EKF (ASEKF) and augmented state UKF (ASUKF) estimators, and NMPC controller.
- to couple NMPC controller with ASEKF and ASUKF in order to formulate two state estimation-based NMPC controllers. That is, to formulate ASEKF-NMPC and ASUKF-NMPC controllers. This means that both ASEKF and ASUKF will be coupled to the same NMPC controller.
- to generate Pareto fronts for ASEKF-NMPC and ASUKF-NMPC controllers using one of the Multiobjective Optimization Evolutionary Algorithms (MOEAs), called Non-dominated Sorting Genetic Algorithm (NSGA-II) algorithm. The Pareto fronts are compared using *Level diagrams* in order to analyze the closed-loop performances of ASEKF and ASUKF.

1.6 Scope and Limitations

There are many different nonlinear optimal state estimation methods that can be implemented in the NMPC formulation, but in this thesis only ASEKF and ASUKF are considered. The resultant algorithms: ASEKF-NMPC and ASUKF-NMPC, are applied for the feedback control of an inverted pendulum on a moving cart system. The evaluation problem is converted into the multiobjective optimization problem and NSGA-II is used to solve it. For visualization and analysis of the resulting Pareto fronts, *Level Diagrams tool* is employed.

1.7 Thesis Outline

The thesis is organized as follows:

- Chapter 1 provides a general background of the whole thesis, research contributions and a problem statement.
- Chapter 2 discusses a concept of dynamic system modeling as it will be used extensively in the following chapters. It discusses specifically about how to convert continuous time dynamic systems to discrete time dynamic systems using numerical integration methods and an inclusion of disturbances in the system model.
- Chapter 3 presents a literature review of optimal state estimation for nonlinear dynamic systems. The state estimation methods of interest for this chapter are the Extended Kalman Filter and Unscented Kalman Filter.
- Chapter 4 provides a literature review of the nonlinear model predictive control. It discusses how NMPC schemes perform state and output predictions over a given prediction horizon. Furthermore, it discusses how NMPC optimal control problem is formulated and how it can be converted into a nonlinear programming problem.
- Chapter 5 covers the design of ASEKF and ASUKF estimators. The ASEKF and ASUKF are then used for the design of ASEKF-NMPC and ASUKF-NMPC controllers.
- Chapter 6 provides a literature review of multiobjective optimization and level diagram representation of Pareto fronts.
- Chapter 7 presents a derivation of a model of an inverted pendulum on a moving cart system. The inverted pendulum system is used as benchmark for the MOO evaluation of ASEKF and ASUKF estimators.
- Chapter 8 presents a procedure which is followed to carry out the MOO evaluation of ASEKF and ASUKF estimators. The evaluation results are also provided in this chapter.
- Chapter 9 draws the conclusions on the work covered in this thesis and makes suggestions for the future work.

Chapter 2

Dynamic System Modeling Review

A concept of dynamic system modeling is fundamental when designing control systems and estimation techniques which form a major body of work of this thesis. In this chapter a concept of dynamic system modeling is introduced as a preparation for discussions which are provided later in this thesis. The chapter begins by providing a general state space framework for dynamic system models, which gives functional mapping between state evolution, and states and inputs of the system. The chapter goes on to introduce numerical integration methods for converting state space models from continuous time domain to discrete time domain. The process of converting systems in continuous time domain into discrete time domain is called *discretization*. It is one of the most critical steps when implementing control and estimation methods on digital computers. The last section presents augmentation method for handling uncertainty in the state space formulation. For other augmentation methods, see [8].

2.1 State Space Representation

Representing physical system models in state space form is beneficial since it makes it simpler to analyze its dynamics. It makes it clear how each state variable evolves with time. It can also easily handle nonlinear and multivariable systems.

A state space formulation represents a physical system model by a set of n first-order ordinary differential equations, in which time derivatives of its state variables are expressed in terms of the state variables and input variables. The result is a general representation of the form

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \tag{2.1}$$

where $\mathbf{x}(t)$ is a *state vector*, $\mathbf{u}(t)$ is a *control input vector* and t is called a *time variable*. A functional mapping $\mathbf{f}$ is called a *state transition function*. It describes how time evolution of state variables relate to the states and input variables of the system.

The physical quantities which can be measured directly from the system are called system outputs [3]. As with time evolution of states given by Equation (2.1), the system output

can as well be expressed as the function of states, but in most cases does not have direct dependence on inputs.

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)), \tag{2.2}$$

where $\mathbf{y}(t)$ is an *output vector* and the function $\mathbf{h}$ represents a set of functions that relate the output variables to the state variables and input variables.

Bringing Equations (2.1) and (2.2) together, a *complete* state space model of the physical system is formed. This means that the model of the system is completely described by a set of n state equations and the set of m output equations that relate output variables to the state and inputs variables. The complete state space model of the system is given by Equations (2.3) and (2.4) as follows

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \tag{2.3}$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)). \tag{2.4}$$

Notice that so far state space model derived above is in continuous-time domain which respects the fact that almost all physical systems which are dealt with in engineering depend continuously on time variable. Having formulated state space model of the system, it is now required that the model be presented in a way that is practically usable. For the purpose of this work, discrete time models are preferred over continuous time models owing to the fact that the models are usually implemented on the digital computers [14].

2.2 Numerical Integration

To transform continuous time model into its discrete time counter-part, a first order time derivative of the state vector on the left hand-side of Equation (2.3) is approximated using various numerical integration methods. There are many numerical integration methods in literature which can be used, but the most widely used are Euler and Runge-Kutta methods. Euler methods are classified as single step methods while Runge-Kutta methods are multistep methods. A difference between two classifications will be apparent by the end of this section.

2.2.1 Euler Methods

Equation (2.5) is called a forward difference formula and it is used for approximating the differential equation given by Equation (2.3) using *forward Euler method* of integration.

$$\frac{d\mathbf{x}(t)}{dt} = \frac{\mathbf{x}(k+1) - \mathbf{x}(k)}{T_s}, \tag{2.5}$$

where $\mathbf{x}(k)$ is a discrete state vector, k represents the discrete instants of the time variable and T_s is called a sampling time. Solving for $\mathbf{x}(k+1)$ and making it a subject of the

formula, Equation (2.5) is rewritten as

$$\begin{aligned}\mathbf{x}(k+1) &= \mathbf{x}(k) + T_s \frac{d\mathbf{x}(t)}{dt}, \\ &= \mathbf{x}(k) + T_s \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)).\end{aligned} \tag{2.6}$$

Equation (2.6) is called forward Euler method. It relates state at the next time step to the state and input at the current time step. In a similar way, a *backward Euler method* is derived from backward difference formula and it is given as

$$\mathbf{x}(k) = \mathbf{x}(k-1) + T_s \mathbf{f}(\mathbf{x}(k-1), \mathbf{u}(k-1)). \tag{2.7}$$

In contrast to forward Euler method, backward Euler method relates the state at the current time step to state and input at the previous time step. Furthermore, a *central Euler method* is given as

$$\mathbf{x}(k+1) = \mathbf{x}(k-1) + 2T_s \mathbf{f}(\mathbf{x}(k-1), \mathbf{u}(k-1)). \tag{2.8}$$

A central Euler method relates the state at the next time step to the state and input at the previous time step. Euler methods are based on Taylor series expansion with a *local truncation error* (LTE) of order $O\left(T_s^2\right)$.

2.2.2 Runge-Kutta Methods

If system dynamics are highly nonlinear or a desired sampling time is large, more accurate approximations are required and Runge-Kutta methods are preferred in such cases. There are many versions of Runge-Kutta methods. For instance, a second-order Runge-Kutta (RK2), a third-order Runge-Kutta (RK3), a forth-order Runge-Kutta (RK4) and many other higher order methods. The order depends on a degree of accuracy desired. The higher the order, the better the approximation, but the larger the computational time.

RK2 is given by Equations (2.9) through (2.11) as follows

$$\mathbf{x}(k+1) = \mathbf{x}(k) + T_s \left(\frac{1}{2}\mathbf{k}_1 + \frac{1}{2}\mathbf{k}_2\right), \tag{2.9}$$

$$\mathbf{k}_1 = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \tag{2.10}$$

$$\mathbf{k}_2 = \mathbf{f}(\mathbf{x}(t) + T_s\mathbf{k}_1, \mathbf{u}(t)), \tag{2.11}$$

where $\mathbf{k}_1$ and $\mathbf{k}_2$ are *stages* of RK2 method. Notice that when $\mathbf{k}_2 = \mathbf{0}$, Equation (2.9) reduces to Equation (2.6). In a similar way, a well-known classical RK4 is given by Equations

(2.12) through (2.16) as follows

$$\mathbf{x}(k+1) = \mathbf{x}(k) + T_s\left(\frac{1}{6}\mathbf{k}_1 + \frac{1}{3}\mathbf{k}_2 + \frac{1}{3}\mathbf{k}_3 + \frac{1}{6}\mathbf{k}_4\right), \tag{2.12}$$

$$\mathbf{k}_1 = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)), \tag{2.13}$$

$$\mathbf{k}_2 = \mathbf{f}\left(\mathbf{x}(t) + \frac{T_s}{2}\mathbf{k}_1, \mathbf{u}(t)\right), \tag{2.14}$$

$$\mathbf{k}_3 = \mathbf{f}\left(\mathbf{x}(t) + \frac{T_s}{2}\mathbf{k}_2, \mathbf{u}(t)\right), \tag{2.15}$$

$$\mathbf{k}_4 = \mathbf{f}(\mathbf{x}(t) + T_s\mathbf{k}_3, \mathbf{u}(t)), \tag{2.16}$$

where $\mathbf{k}_1$, $\mathbf{k}_2$, $\mathbf{k}_3$ and $\mathbf{k}_4$ are known as stages of RK4 method. When $\mathbf{k}_2 = \mathbf{k}_3 = \mathbf{k}_4 = \mathbf{0}$ Equation (2.12) reduces down to Equation (2.5).

RK2 and RK4 have LTEs of order $O\left(T_s^3\right)$ and $O\left(T_s^5\right)$, respectively. Runge-Kutta methods of higher order can be developed in a similar fashion. The complete discrete time state space model of the physical system described by Equations (2.3) and (2.4) is therefore, given by

$$\mathbf{x}(k) = \mathbf{f}(\mathbf{x}(k-1), \mathbf{u}(k-1)), \tag{2.17}$$

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{x}(k)), \tag{2.18}$$

where $\mathbf{y}(k)$ is a discrete output vector and k is a discrete time variable. Equation (2.17) is usually derived from Euler methods, but if an approximation accuracy is of main interest, then the first-order Euler method can be replaced by any higher-order Runge-Kutta method, such as RK4.

2.3 Including Disturbances

The discussions provided in the previous sections ignore the fact that almost all dynamic systems are prone to unknown disturbances. It is important to discuss and develop a good understand on how disturbances and plant-model mismatches affect the closed-loop behavior of the system and how they can be incorporated in system model in order to suppress their effects.

Disturbances can be modeled and their models included in the state space model and then estimated using state estimation methods [8, 14]. Through out this thesis, augmentation method is adopted for handling disturbances and plant-model mismatches in state space framework. In this method, disturbance dynamics are appended to system dynamics framework and treated as additional dynamics of the system. Literature shows that this method is important when designing a robust MPC for the systems which are prone to disturbances [5, 8, 14]. A disadvantage of this method is that it increases dimensionality of the state space model, hence increasing its computational burden.

The unknown disturbances can occur at the input and output of the system and they

are called *input* $\mathbf{v}(k-1)$ and *output* $\mathbf{d}(k)$ disturbances, respectively. Mathematically speaking, an incorporation of input and output disturbances into the state space model is done by adding $\mathbf{v}(k-1)$ and $\mathbf{d}(k)$ to Equations (2.17) and (2.18), respectively, in the following manner

$$\mathbf{x}(k) = \mathbf{f}(\mathbf{x}(k-1), (\mathbf{u}(k-1) + \Gamma_{\mathbf{v}}\mathbf{v}(k-1))), \tag{2.19}$$

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{x}(k)) + \Gamma_{\mathbf{d}}\mathbf{d}(k). \tag{2.20}$$

where $\Gamma_{\mathbf{v}}$ and Γ_d are the weighting matrices determining significance of the disturbances in the system. From these equations, disturbances come in as additive terms, but in more practical situations they may not necessarily be additive quantities. That is shown by passing $\mathbf{v}(k-1)$ and $\mathbf{d}(k)$ as arguments of Equations (2.17) and (2.18), respectively. Equations (2.17) and (2.18) are special cases of Equations (2.19) and (2.20), respectively since they occur only when $\Gamma_{\mathbf{v}} = \mathbf{0}^{(n\times n)}$ and $\Gamma_{\mathbf{d}} = \mathbf{0}^{(m\times m)}$.

2.3.1 Output Disturbances

For the sake of simplicity, the author assumes that output disturbances are additive to output variables as it has already been mentioned above. To simplify further, the author also assumes that output disturbance can be modeled by a step function of a given magnitude. Therefore, $\mathbf{d}(k)$ is modeled as

$$\mathbf{d}(k) = \mathbf{d}(k-1). \tag{2.21}$$

It is now convenient to cast disturbance model given by Equation (2.21) into a single framework with the system model given by Equations (2.19) and (2.20) to allow us to form a new state space model and design the state estimation method for it.

$$\begin{bmatrix} \mathbf{x}(k) \\ \mathbf{d}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\mathbf{x}(k-1), \mathbf{u}(k-1)) \\ \mathbf{d}(k-1) \end{bmatrix}, \tag{2.22}$$

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{x}(k)) + \Gamma_{\mathbf{d}}\mathbf{d}(k). \tag{2.23}$$

Note from this model that the variables in $\mathbf{d}(k)$ are treated as additional state variables of the system. Since $\mathbf{d}(k)$ is unknown, the model can be used to simultaneously estimate both system states and output disturbances. Therefore, the new state vector is defined as

$$\mathbf{x}_{\mathbf{a}}(k) = \begin{bmatrix} \mathbf{x}^T(k) & \mathbf{d}^T(k) \end{bmatrix}^T \tag{2.24}$$

where $\mathbf{x}_{\mathbf{a}}(k)$ is called the augmented state vector.

2.3.2 Input Disturbances

As with the output disturbance, input disturbance can be represented by the step function of the given magnitude. Therefore, $\mathbf{v}(k)$ can be modeled as

$$\mathbf{v}(k+1) = \mathbf{v}(k) \tag{2.25}$$

In the similar manner, input disturbance model described by Equation (2.25) is augmented into the state space model to create a state space model which involves input disturbance dynamics as part of the system dynamics. From this new formulation, the state estimation methods can be implemented to jointly estimate input disturbance and system states. Designing state estimation methods is the subject of Chapter 3, hence it is not carried out here.

$$\begin{bmatrix} \mathbf{x}(k) \\ \mathbf{v}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\mathbf{x}(k-1), (\mathbf{u}(k-1) + \Gamma_v \mathbf{v}(k-1))) \\ \mathbf{v}(k-1) \end{bmatrix}, \tag{2.26}$$

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{x}(k)). \tag{2.27}$$

The resultant model can be used to design a controller with an input disturbance rejection property, whereby the controller is able to suppress disturbances. The new state vector is given by

$$\mathbf{x_a}(k) = \begin{bmatrix} \mathbf{x}^T(k) & \mathbf{v}^T(k) \end{bmatrix}^T. \tag{2.28}$$

2.3.3 Plant-Model Mismatches

Generally, a mismatch (whether structural or parametric) often exists between the actual plant dynamics and the model which represents those dynamics in a mathematical setting. This mismatch is called plant-model mismatch. Plant-model mismatches are known to cause steady-state offsets in control systems, but there are various ways in literature which have been proposed to handle them. In this thesis we are interested in model augmentation in which mismatch uncertainty is modeled and augmented into the dynamic model. This thesis focuses specifically on *parametric* plant-model mismatch in which parameters of the system are not fully known.

Representing a vector containing unknown system parameters by $\mathbf{w}(k)$, the joint state vector of the system states and parameters is given by

$$\mathbf{x_a}(k) = \begin{bmatrix} \mathbf{x}^T(k) & \mathbf{w}^T(k) \end{bmatrix}^T \tag{2.29}$$

and the corresponding state space model is reformulated as

$$\begin{bmatrix} \mathbf{x}(k) \\ \mathbf{w}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\mathbf{x}(k-1), \mathbf{u}(k-1), \mathbf{w}(k-1)) \\ \mathbf{w}(k-1) \end{bmatrix}, \tag{2.30}$$

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{x}(k), \mathbf{w}(k)). \tag{2.31}$$

This model can be implemented in the design of a state estimation method to simultaneously estimate states and parameters of the system. This is called *joint filtering approach*. Another approach which is usually used in most applications is called *dual filtering approach*. It is different from joint filtering approach in that both state and parameters are no longer estimated simultaneously. Rather, parameter estimation is carried out separate from state estimation. This is illustrated in Figure 2.1. The left-hand part of the figure il-

lustrates the joint filtering approach and right-hand part illustrates dual filtering approach. The vectors $\mathbf{y}_k$ and $\mathbf{w}_k$ are similar to $\mathbf{y}(k)$ and $\mathbf{w}(k)$, respectively, while $\hat{\mathbf{x}}_k$ and $\hat{\mathbf{w}}_k$ are the estimates of $\mathbf{y}_k$ and $\mathbf{w}_k$, respectively.

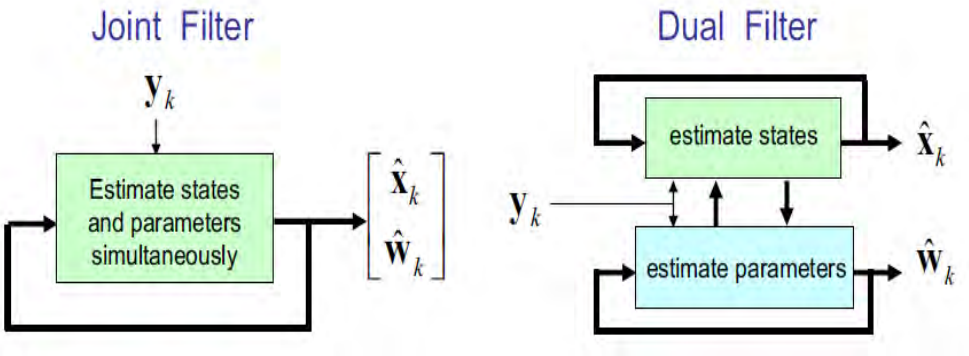


Figure 2.1: Dual and joint estimation approaches [18].

The total model augmentation which takes into account both input and output disturbances and plant-model mismatch uncertainties is given by the following equations,

$$\begin{bmatrix} \mathbf{x}(k) \\ \mathbf{v}(k) \\ \mathbf{d}(k) \\ \mathbf{w}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\mathbf{x}(k-1), (\mathbf{u}(k-1) + \Gamma_{\mathbf{v}}\mathbf{v}(k-1)), \mathbf{w}(k-1)) \\ \mathbf{v}(k-1) \\ \mathbf{d}(k-1) \\ \mathbf{w}(k-1) \end{bmatrix}, \tag{2.32}$$

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{x}(k), \mathbf{w}(k)) + \Gamma_{\mathbf{d}}\mathbf{d}(k). \tag{2.33}$$

In Equation (2.32), disturbances and parameters are now seen as additional states of the system. That is, they are now casted into one big state vector. Notice that Equation (2.33) is still similar to Equation (2.18) in terms of the size of $\mathbf{y}(k)$ since the augmentation process does not affect the number of system outputs. The new state vector is given by

$$\mathbf{x}_{\mathbf{a}}(k) = \begin{bmatrix} \mathbf{x}^T(k) & \mathbf{v}^T(k) & \mathbf{d}^T(k) & \mathbf{w}^T(k) \end{bmatrix}^T. \tag{2.34}$$

Chapter 3

Nonlinear State Estimation

In this chapter, a short review of state estimation theory is given. The review introduces some concepts which are fundamental to the formulation of optimal state estimation methods. Later on in this chapter, two well-known nonlinear optimal state estimation methods: UKF and EKF are presented. There are many texts which explain the fundamental concepts of state estimation in much more details and they are cited throughout this chapter. For instance, Simon in [19] gives a good literature on optimal state estimation theory and explains formulations of various state estimation methods. Van der Merwe in [18] provides a detailed literature on sigma point Kalman filtering. Hence, most of the material covered in this chapter is taken from [18, 19].

3.1 Introduction

State estimation is important for model-based control applications which require full-state feedback. For instance, in order to implement Model Predictive Control (MPC) or Linear Quadratic Regulator (LQR) schemes, it is required that all state variables of a controlled system be measurable. Otherwise, state estimation methods can be implemented to compute the estimates of unmeasurable states using system dynamics and noisy measurements [19, 22]. The amount of measurements available for state estimation varies from one application to another. If all measurements up to and including time instance k are available, then *a posteriori* estimate is formed [19].

It is a common practice in many texts of recursive state estimation to begin by introducing a *linear* Kalman Filter (KF). A reason is that KF provides a framework from which many other advanced state estimation methods can be derived. KF was invented by Rudolf Emil Kalman in 1960s as part of Apollo program, which required the estimates of trajectories of a manned spacecraft going to the moon and back to earth [23]. Although the initial uses of KF were in aerospace applications, it has been adapted for use in many other applications such as consumer, health, commercial and defense applications [23].

Linear Kalman filter can be used only in cases where the system has linear dynamics, but once the dynamics are nonlinear, other advanced state estimation methods have been

developed to handle it. EKF is the most widely employed state estimation method for nonlinear estimation problems. It works by propagating the pdf of state estimates through a *linear approximation* of the system model around an operating point and then applies KF equations. In practice, however, linear approximation may introduce large estimation errors. In order words, with this method the state estimation method may be unstable, especially if the system is highly nonlinear [13]. On the other hand, UKF propagates the pdf of the state estimates through a nonlinear transformation called *Unscented Transformation*. It is alternative way of propagating the pdf of the state estimates through nonlinearity without having to compute Jacobians or Hessians. Literature suggests that UKF has shown better accuracy performance in many estimation applications compared to EKF because it employes unscented transformation.

3.2 Kalman Filtering Framework

The basic framework for the whole of Kalman filtering is linear Kalman filter. It involves estimation of the states of discrete time linear dynamic system described by Equations (3.1) and (3.2).

$$\mathbf{x}(k) = \mathbf{A}\mathbf{x}(k-1) + \mathbf{B}\mathbf{u}(k-1) + \vartheta(k-1), \tag{3.1}$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \xi(k), \tag{3.2}$$

where $\mathbf{A}$ is a state transition matrix, $\mathbf{B}$ is an input matrix and $\mathbf{C}$ is the output matrix. The process and measurement noise terms, $\vartheta(k)$ and $\xi(k)$, respectively, are assumed to be Gaussian white, zero-mean, uncorrelated and they have known covariance matrices $\mathbf{Q}(k)$ and $\mathbf{R}(k)$, respectively. That is

$$\vartheta(k) \sim N(\mathbf{0}, \mathbf{Q}(k)), \tag{3.3}$$

$$\xi(k) \sim N(\mathbf{0}, \mathbf{R}(k)), \tag{3.4}$$

$$E\left[\vartheta(k)\,\vartheta(k)^T\right] = \mathbf{Q}(k)\,\delta(k-i), \tag{3.5}$$

$$E\left[\xi(k)\,\xi(k)^T\right] = \mathbf{R}(k)\,\delta(k-i), \tag{3.6}$$

$$E\left[\vartheta(k)\,\xi(k)^T\right] = E\left[\xi(k)\,\vartheta(k)^T\right] = 0, \tag{3.7}$$

where N represents a normal distribution, E is an expected value, where $\delta(k-i)$ is the Kronecker delta function, that is, $\delta(k-i) = 1$ if $k = i$ and $\delta(k-i) = 0$ if $k \neq i$.

3.2.1 Linear Kalman Filter

Given the dynamic system described by (3.1) and (3.2), a linear Kalman filter is applied to estimate the current state vector $\mathbf{x}(k)$. It is applied in two phases: prediction (or time-update step) and update (or measurement-update step) [24]. The prediction phase is made up of Equations (3.8) and (3.9). In the prediction phase, the filter uses previous state estimate $\hat{\mathbf{x}}(k-1)$ and previous estimation error covariance $\mathbf{P}_{\mathbf{x}}(k-1)$ to compute the predicted estimates of the state and estimation error covariance, $\hat{\mathbf{x}}^{-}(k)$ and $\mathbf{P}_{\mathbf{x}}^{-}(k)$,

respectively

$$\hat{\mathbf{x}}^{-}(k) = \mathbf{A}\hat{\mathbf{x}}(k-1) + \mathbf{B}\mathbf{u}(k-1), \tag{3.8}$$

$$\mathbf{P}_{\mathbf{x}}^{-}(k) = \mathbf{A}\mathbf{P}_{\mathbf{x}}(k-1)\mathbf{A}^{T} + \mathbf{Q}(k-1). \tag{3.9}$$

The measurement update phase is made up of Equations (3.10) through (3.12). In the measurement update phase, the filter updates $\hat{\mathbf{x}}^{-}(k)$ and $\mathbf{P}_{\mathbf{x}}^{-}(k)$ using $\mathbf{y}(k)$ in order to refine $\hat{\mathbf{x}}^{-}(k)$ and $\mathbf{P}_{\mathbf{x}}^{-}(k)$ into *a posteriori* estimates. The filter achieves *a posteriori* estimates of the state and error covariance, $\hat{\mathbf{x}}(k)$ and $\mathbf{P}_{\mathbf{x}}(k)$, respectively, by using an optimal Kalman gain $\mathbf{K}(k)$ calculated using Equation (3.10).

$$\mathbf{K}(k) = \mathbf{P}_{\mathbf{x}}^{-}(k)\mathbf{C}^{T}\left(\mathbf{C}\mathbf{P}_{\mathbf{x}}^{-}(k)\mathbf{C}^{T} + \mathbf{R}(k)\right)^{-1}, \tag{3.10}$$

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}^{-}(k) + \mathbf{K}(k)\left(\mathbf{y}(k) - \mathbf{C}\hat{\mathbf{x}}^{-}(k)\right), \tag{3.11}$$

$$\mathbf{P}_{\mathbf{x}}(k) = \left(\mathbf{I} - \mathbf{K}(k)\mathbf{C}\right)\mathbf{P}_{\mathbf{x}}^{-}(k), \tag{3.12}$$

where $\mathbf{I}$ is an identity matrix. A "-" symbol denotes *a priori* estimate. The *a priori* and *a posteriori* mean before a measurement is taken and after the measurement is taken, respectively. Algorithm 3.1 provides an implementable linear Kalman filter algorithm.

Algorithm 3.1 Implementable linear Kalman filter algorithm.

- Initialization at $k = 0$:

$$\hat{\mathbf{x}}(0) = E[\mathbf{x}(0)], \tag{3.13}$$

$$\mathbf{P}_{\mathbf{x}}(0) = E\left[\left(\mathbf{x}(0) - \hat{\mathbf{x}}(0)\right)\left(\mathbf{x}(0) - \hat{\mathbf{x}}(0)\right)^{T}\right]. \tag{3.14}$$

- For $k = 1, \cdots, \infty$:

  1. Time-update equations

$$\hat{\mathbf{x}}^{-}(k) = \mathbf{A}\hat{\mathbf{x}}(k-1) + \mathbf{B}\mathbf{u}(k-1), \tag{3.15}$$

$$\mathbf{P}_{\mathbf{x}}^{-}(k) = \mathbf{A}\mathbf{P}_{\mathbf{x}}(k-1)\mathbf{A}^{T} + \mathbf{Q}(k-1). \tag{3.16}$$

  2. Measurement-update equations

$$\mathbf{K}(k) = \mathbf{P}_{\mathbf{x}}^{-}(k)\mathbf{C}^{T}\left(\mathbf{C}\mathbf{P}_{\mathbf{x}}^{-}(k)\mathbf{C}^{T} + \mathbf{R}(k)\right)^{-1}, \tag{3.17}$$

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}^{-}(k) + \mathbf{K}(k)\left(\mathbf{y}(k) - \mathbf{C}\hat{\mathbf{x}}^{-}(k)\right), \tag{3.18}$$

$$\mathbf{P}_{\mathbf{x}}(k) = \left(\mathbf{I} - \mathbf{K}(k)\mathbf{C}\right)\mathbf{P}_{\mathbf{x}}^{-}(k). \tag{3.19}$$

3.2.2 Nonlinear Systems

Applicability of linear Kalman filter is limited to systems whose dynamics are linear. If the system is nonlinear, then nonlinear state estimation methods have to be implemented. The basic framework of nonlinear state estimation methods involves estimation of the state of discrete time nonlinear system described by Equations (3.20) and (3.21) as follows,

$$\mathbf{x}(k) = \mathbf{f}\left(\mathbf{x}(k-1), \mathbf{u}(k-1)\right) + \vartheta(k-1), \tag{3.20}$$

$$\mathbf{y}(k) = \mathbf{h}\left(\mathbf{x}(k)\right) + \xi(k). \tag{3.21}$$

Notice that Equations (3.20) and (3.21) are special cases of more general Equations (3.22) and (3.23), respectively, because they assume $\vartheta(k-1)$ and $\xi(k)$ to be additive.

$$\mathbf{x}(k) = \mathbf{f}(\mathbf{x}(k-1), \mathbf{u}(k-1), \vartheta(k-1)), \tag{3.22}$$

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{x}(k), \xi(k)). \tag{3.23}$$

In Equations (3.22) and (3.23), the process and measurement noise terms are not assumed to be additive, hence why they are passed as parameters of $\mathbf{f}$ and $\mathbf{h}$.

The simple block diagrams representing nonlinear state estimation problems for nonlinear dynamic systems given by Equations (3.20) and (3.21), and Equations (3.22) and (3.23) are shown in Figures 3.1 and 3.2, respectively. Figure 3.1 shows a simplified case of Figure 3.2, where process and measurement noise terms are assumed to be additive. These figures help to explain different formulations of EKF and UKF in Sections 3.3 and 3.4 based on whether process and measurement noise terms are additive or non-additive.

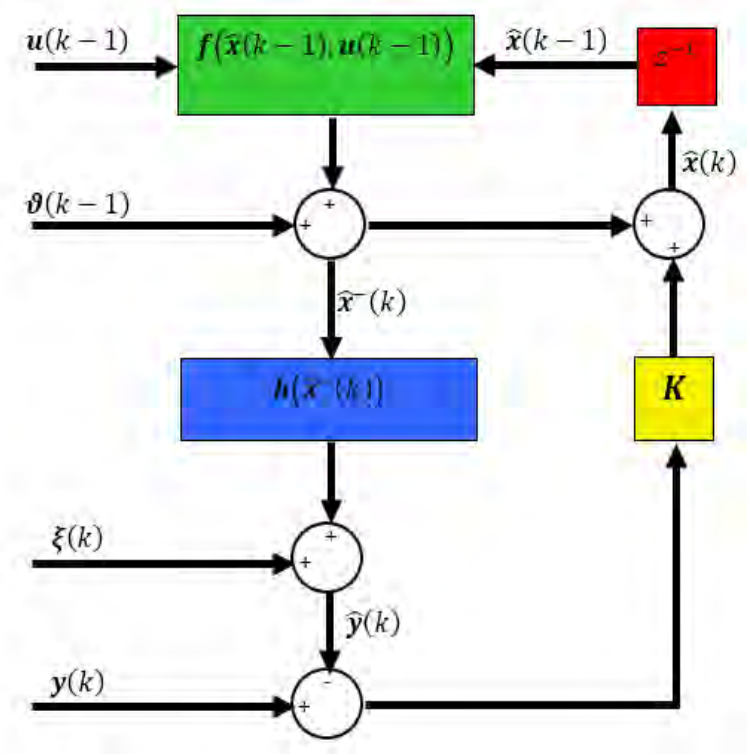


Figure 3.1: Conceptual Representation of Additive Noise for Nonlinear State Estimation.

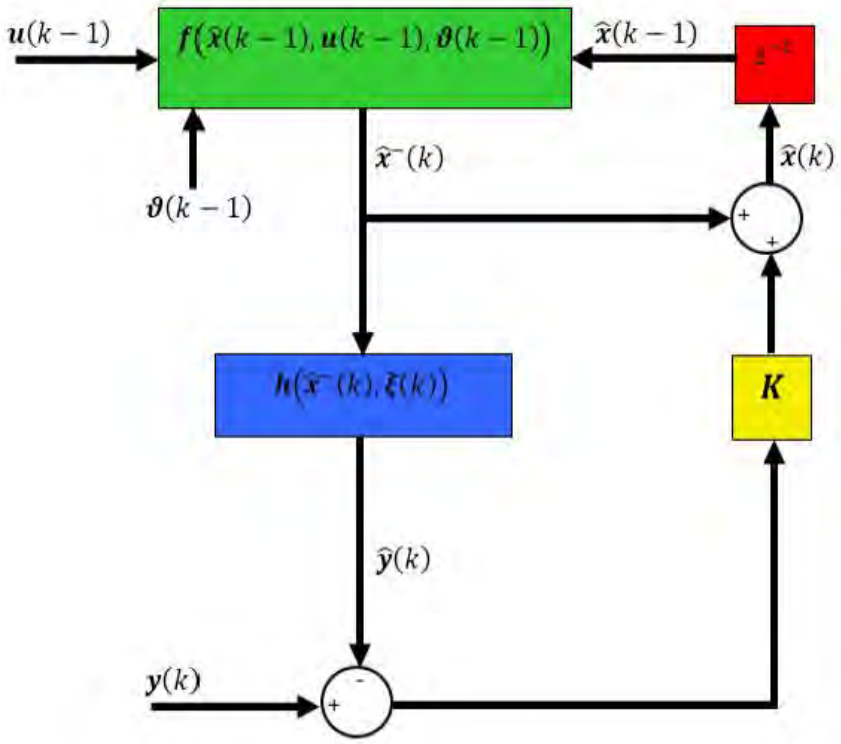


Figure 3.2: Conceptual Representation of Non-Additive Noise for Nonlinear State Estimation.

3.3 Extended Kalman Filter

EKF inherits most of the essential properties of linear KF, but the only difference is that EKF computes Jacobian matrices of the system model at each time step to determine the linearized system model. Once Jacobians are computed, linear KF equations are applied to the linearized model to determine the EKF state estimates and the estimation error covariances.

3.3.1 Linear Approximation

EKF approximates the state distribution by Gaussian random variable (GRV). The GRV is transformed analytically through first-order linearization of nonlinear dynamic system at each time instant [15]. Linearization is known to introduce cumulative errors in the true *a posteriori* mean and covariance of the transformed GRV, which may lead to suboptimal performance of the EKF and sometimes divergence of the EKF [15].

Since the idea of linearizing nonlinear dynamic system forms the critical part of EKF formulation, it is important to discuss how EKF performs linearization. Consider the general nonlinear dynamic system represented by Equations (3.22) and (3.23) which are rewritten here as

$$\mathbf{x}(k) = \mathbf{f}(\mathbf{x}(k-1), \mathbf{u}(k-1), \vartheta(k-1)), \tag{3.24}$$

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{x}(k), \xi(k)). \tag{3.25}$$

It is assumed that the functions $\mathbf{f}$ and $\mathbf{h}$ are sufficiently differentiable and continuous in $\mathbf{x}$, ϑ and ξ so that each one has a valid Taylor series expansion. The process model Jacobians are computed as follows

$$\mathbf{J}_{\mathbf{f}_\mathbf{x}} = \nabla_\mathbf{x}\mathbf{f}\left(\mathbf{x},\, \mathbf{u}\left(k-1\right),\, \vartheta\left(k-1\right)\right)|_{\mathbf{x}=\hat{\mathbf{x}}(k-1)} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{\mathbf{x}=\hat{\mathbf{x}}(k-1)}, \tag{3.26}$$

$$\mathbf{J}_{\mathbf{f}_\vartheta} = \nabla_\vartheta\mathbf{f}\left(\hat{\mathbf{x}}\left(k-1\right),\, \mathbf{u}\left(k-1\right),\, \vartheta\right)|_{\vartheta=\vartheta(k-1)} = \begin{bmatrix} \frac{\partial f_1}{\partial \vartheta_1} & \frac{\partial f_1}{\partial \vartheta_2} & \cdots & \frac{\partial f_1}{\partial \vartheta_n} \\ \frac{\partial f_2}{\partial \vartheta_1} & \frac{\partial f_2}{\partial \vartheta_2} & \cdots & \frac{\partial f_2}{\partial \vartheta_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial \vartheta_1} & \frac{\partial f_n}{\partial \vartheta_2} & \cdots & \frac{\partial f_n}{\partial \vartheta_n} \end{bmatrix}_{\vartheta=\vartheta(k-1)}. \tag{3.27}$$

In the similar manner, the measurement model Jacobians are computed as follows

$$\mathbf{J}_{\mathbf{h}_\mathbf{x}} = \nabla_\mathbf{x}\mathbf{h}\left(\mathbf{x},\, \xi\left(k\right)\right)|_{\mathbf{x}=\hat{\mathbf{x}}^-(k)} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \cdots & \frac{\partial h_2}{\partial x_n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial h_n}{\partial x_1} & \frac{\partial h_n}{\partial x_2} & \cdots & \frac{\partial h_n}{\partial x_n} \end{bmatrix}_{\mathbf{x}=\hat{\mathbf{x}}^-(k)}, \tag{3.28}$$

$$\mathbf{J}_{\mathbf{h}_\xi} = \nabla_\xi\mathbf{h}\left(\hat{\mathbf{x}}^-\left(k\right),\, \xi\right)|_{\xi=\xi(k)} = \begin{bmatrix} \frac{\partial h_1}{\partial \xi_1} & \frac{\partial h_1}{\partial \xi_2} & \cdots & \frac{\partial h_1}{\partial \xi_m} \\ \frac{\partial h_2}{\partial \xi_1} & \frac{\partial h_2}{\partial \xi_2} & \cdots & \frac{\partial h_2}{\partial \xi_m} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial h_m}{\partial \xi_1} & \frac{\partial h_m}{\partial \xi_2} & \cdots & \frac{\partial h_m}{\partial \xi_m} \end{bmatrix}_{\xi=\xi(k)}. \tag{3.29}$$

3.3.2 Implementation Variations

An EKF algorithm for systems with additive process and measurement noise terms is shown in Algorithm 3.2. The more general case in which the process and measurement noise terms are not considered to be necessarily additive is shown in Algorithm 3.3. In this case, Equations (3.27) and (3.29) are necessary and they have to be used in the EKF formulation. Equations (3.26) and (3.28) are used to compute Jacobian matrices, $\mathbf{J}_{\mathbf{f}_\mathbf{x}}$ and $\mathbf{J}_{\mathbf{h}_\mathbf{x}}$, respectively, in Algorithm 3.2.

Algorithm 3.2 Extended Kalman Filter (EKF) Algorithm - Additive Noise.

- Initialization at $k = 0$

$$\hat{\mathbf{x}}(0) = E[\mathbf{x}(0)], \tag{3.30}$$

$$\mathbf{P}_{\mathbf{x}}(0) = E\left[(\mathbf{x}(0) - \hat{\mathbf{x}}(0))(\mathbf{x}(0) - \hat{\mathbf{x}}(0))^T\right]. \tag{3.31}$$

- For $k = 1, \cdots, \infty$:

  1. Time-update equations:

$$\mathbf{J}_{\mathbf{f}_{\mathbf{x}}} = \nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x}, \mathbf{u}(k-1))|_{\mathbf{x}=\hat{\mathbf{x}}(k-1)}, \tag{3.32}$$

$$\hat{\mathbf{x}}^-(k) = \mathbf{f}(\hat{\mathbf{x}}(k-1), \mathbf{u}(k-1)), \tag{3.33}$$

$$\mathbf{P}_{\mathbf{x}}^-(k) = \mathbf{J}_{\mathbf{f}_{\mathbf{x}}}\mathbf{P}_{\mathbf{x}}(k-1)\mathbf{J}_{\mathbf{f}_{\mathbf{x}}}{}^T + \mathbf{Q}. \tag{3.34}$$

  2. Measurement-update equations:

$$\mathbf{J}_{\mathbf{h}_{\mathbf{x}}} = \nabla_{\mathbf{x}}\mathbf{h}(\mathbf{x})|_{\mathbf{x}=\hat{\mathbf{x}}^-(k)}, \tag{3.35}$$

$$\mathbf{K}(k) = \mathbf{P}_{\mathbf{x}}^-(k)\mathbf{J}_{\mathbf{h}_{\mathbf{x}}}{}^T\left[\mathbf{J}_{\mathbf{h}_{\mathbf{x}}}\mathbf{P}_{\mathbf{x}}^-(k)\mathbf{J}_{\mathbf{h}_{\mathbf{x}}}{}^T + \mathbf{R}\right]^{-1}, \tag{3.36}$$

$$\hat{\mathbf{y}}^-(k) = \mathbf{h}\left(\hat{\mathbf{x}}^-(k)\right), \tag{3.37}$$

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}^-(k) + \mathbf{K}(k)\left[\mathbf{y}(k) - \hat{\mathbf{y}}^-(k)\right], \tag{3.38}$$

$$\mathbf{P}_{\mathbf{x}}(k) = \left[\mathbf{I} - \mathbf{K}(k)\mathbf{J}_{\mathbf{h}_{\mathbf{x}}}\right]\mathbf{P}_{\mathbf{x}}^-(k). \tag{3.39}$$

Algorithm 3.3 Extended Kalman Filter (EKF) Algorithm - Non-Additive Noise.

- Initialization at $k = 0$

$$\hat{\mathbf{x}}(0) = E[\mathbf{x}(0)], \tag{3.40}$$

$$\mathbf{P}_{\mathbf{x}}(0) = E\left[(\mathbf{x}(0) - \hat{\mathbf{x}}(0))(\mathbf{x}(0) - \hat{\mathbf{x}}(0))^T\right]. \tag{3.41}$$

- For $k = 1, \cdots, \infty$:

  1. Time-update equations:

$$\mathbf{J}_{\mathbf{f}_{\mathbf{x}}} = \nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x}, \mathbf{u}(k-1), \vartheta(k-1))|_{\mathbf{x}=\hat{\mathbf{x}}(k-1)}, \tag{3.42}$$

$$\mathbf{J}_{\mathbf{f}_{\vartheta}} = \nabla_{\vartheta}\mathbf{f}(\hat{\mathbf{x}}(k-1), \mathbf{u}(k-1), \vartheta)|_{\vartheta=\vartheta(k-1)}, \tag{3.43}$$

$$\hat{\mathbf{x}}^-(k) = \mathbf{f}(\hat{\mathbf{x}}(k-1), \mathbf{u}(k-1), \vartheta(k-1)), \tag{3.44}$$

$$\mathbf{P}_{\mathbf{x}}^-(k) = \mathbf{J}_{\mathbf{f}_{\mathbf{x}}}\mathbf{P}_{\mathbf{x}}(k-1)\mathbf{J}_{\mathbf{f}_{\mathbf{x}}}{}^T + \mathbf{J}_{\mathbf{f}_{\vartheta}}\mathbf{Q}\mathbf{J}_{\mathbf{f}_{\vartheta}}{}^T. \tag{3.45}$$

  2. Measurement-update equations:

$$\mathbf{J}_{\mathbf{h}_{\mathbf{x}}} = \nabla_{\mathbf{x}}\mathbf{h}(\mathbf{x}, \xi(k))|_{\mathbf{x}=\hat{\mathbf{x}}^-(k)}, \tag{3.46}$$

$$\mathbf{J}_{\mathbf{h}_{\xi}} = \nabla_{\xi}\mathbf{h}\left(\hat{\mathbf{x}}^-(k), \xi\right)|_{\xi=\xi(k)}, \tag{3.47}$$

$$\mathbf{K}(k) = \mathbf{P}_{\mathbf{x}}{}^-(k)\mathbf{J}_{\mathbf{h}_{\mathbf{x}}}{}^T\left[\mathbf{J}_{\mathbf{h}_{\mathbf{x}}}\mathbf{P}_{\mathbf{x}}{}^-(k)\mathbf{J}_{\mathbf{h}_{\mathbf{x}}}{}^T + \mathbf{J}_{\mathbf{h}_{\xi}}\mathbf{R}\mathbf{J}_{\mathbf{h}_{\xi}}{}^T\right]^{-1}, \tag{3.48}$$

$$\hat{\mathbf{y}}^-(k) = \mathbf{h}\left(\hat{\mathbf{x}}^-(k), \xi(k)\right), \tag{3.49}$$

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}^-(k) + \mathbf{K}(k)\left[\mathbf{y}(k) - \hat{\mathbf{y}}^-(k)\right], \tag{3.50}$$

$$\mathbf{P}_{\mathbf{x}}(k) = \left[\mathbf{I} - \mathbf{K}(k)\mathbf{J}_{\mathbf{h}_{\mathbf{x}}}\right]\mathbf{P}_{\mathbf{x}}{}^-(k). \tag{3.51}$$

3.4 Unscented Kalman Filter

Tuning EKF can be a difficult process and sometimes result in inaccurate estimates, especially if the dynamic system is highly nonlinear [15, 18, 19]. This is because EKF relies on computing Jacobian matrices to transform the mean and covariance of the state [19]. Unscented Kalman filter (UKF) is a *derivative-free* alternative to EKF. It works by selecting a minimal set of points called sigma points to capture the true mean and covariance of the state estimate. The sigma points are then propagated using unscented transformation (UT) to calculate the mean and covariance of the transformed RV. UT eliminates a difficulty in the derivation and evaluation of Jacobian matrices, hence making UKF much easier to implement and provide improved results compared to EKF in most cases [25].

3.4.1 Unscented Transformation

Unscented Kalman filtering is entirely based on UT technique. UT is the method of calculating the mean and covariance of RV which under goes nonlinear transformation [18]. It is based on two principles. First, it is easier to perform nonlinear transformation on a single point rather than on the entire pdf. Second, it is not too complicated to determine a set of points in state space whose sample pdf approximates the true pdf of the state [19].

Consider propagating the random variable $\mathbf{x}(k)$ through a nonlinear function given by

$$\mathbf{y}(k) = \mathbf{g}(\mathbf{x}(k)). \tag{3.52}$$

Assuming that $\mathbf{x}(k)$ has a mean $\bar{\mathbf{x}}$ and covariance $\mathbf{P}_{\mathbf{x}}$, the question is: how does the UT calculate the mean $\bar{\mathbf{y}}$ and covariance $\mathbf{P}_{\mathbf{y}}$ of the transformed random variable $\mathbf{y}(k)$? To calculate $\bar{\mathbf{y}}$ and $\mathbf{P}_{\mathbf{y}}$, UT selects a set of $2n+1$ sigma points around $\bar{\mathbf{x}}$ and form a sigma point matrix $\mathcal{X}$ as follows

$$\mathcal{X}_0 = \bar{\mathbf{x}}, \quad i = 0, \tag{3.53}$$

$$\mathcal{X}_i = \bar{\mathbf{x}} + \left(\sqrt{(n+\lambda)\mathbf{P}_{\mathbf{x}}(k)}\right)_i, \quad i = 1, \cdots, n, \tag{3.54}$$

$$\mathcal{X}_i = \bar{\mathbf{x}} - \left(\sqrt{(n+\lambda)\mathbf{P}_{\mathbf{x}}(k)}\right)_i, \quad i = n+1, \cdots, 2n, \tag{3.55}$$

where $\lambda = \alpha^2[n+\kappa] - n$ is a scaling parameter. The primary scaling parameter α affects the spread of the sigma points around $\bar{\mathbf{x}}$. This parameter takes any value between 10^{-04} and 1. Smaller α leads to a closer selection of sigma points to $\bar{\mathbf{x}}$, while larger α leads to a wider spread of sigma points from $\bar{\mathbf{x}}$. A tertiary scaling parameter κ is responsible for ensuring the positive semi-definiteness of the covariance matrix $\mathbf{P}_{\mathbf{x}}$. This parameters is usually set to 0. $\left(\sqrt{(n+\lambda)\mathbf{P}_{\mathbf{x}}(k)}\right)_i$ is the ith column of the matrix square root of the weighted covariance matrix, $(n+\lambda)\mathbf{P}_{\mathbf{x}}(k)$. The numerically efficient *Cholesky factorization* method is often used to calculate the matrix square root.

Each sigma point is now propagated through the nonlinear function described by Equation

(3.52) as follows,

$$\mathcal{Y}_i = \mathbf{g}\left(\mathcal{X}_i\right), \quad i = 0, \cdots, 2n. \tag{3.56}$$

The approximated mean and covariance of $\mathbf{y}$ are computed as the weighted average and the weighted outer product of the transformed sigma points $\mathcal{Y}_i$, respectively,

$$\bar{\mathbf{y}} = \sum_{i=0}^{2n} \eta_i^m \mathcal{Y}_i \tag{3.57}$$

$$\mathbf{P}_{\mathbf{y}} = \sum_{i=0}^{2n} \eta_i^c \left(\mathcal{Y}_i - \bar{\mathbf{y}}\right)\left(\mathcal{Y}_i - \bar{\mathbf{y}}\right)^T. \tag{3.58}$$

The approximated cross-covariance of $\mathbf{x}$ and $\mathbf{y}$ can be computed as the weighted outer product $\mathcal{X}_i$ and $\mathcal{Y}_i$ as follows,

$$\mathbf{P}_{\mathbf{xy}} = \sum_{i=0}^{2n} \eta_i^c \left(\mathcal{X}_i - \bar{\mathbf{x}}\right)\left(\mathcal{Y}_i - \bar{\mathbf{y}}\right)^T, \tag{3.59}$$

with the weighting vectors η^m and η^c defined as follows

$$\eta_0^m = \frac{\lambda}{\lambda + n} \tag{3.60}$$

$$\eta_0^m = \frac{\lambda}{\lambda + n} + 1 - \alpha^2 + \beta \tag{3.61}$$

$$\eta_i^m = \eta_i^c = \frac{1}{2\left[n + \lambda\right]}, \; i = 1, \cdots, 2n \tag{3.62}$$

A secondary scaling parameter β is responsible for making sure that the information about *a priori* distribution is included. This parameter takes a default value of 2 which is considered to be optimal.

3.4.2 Implementation Variations

The UKF algorithm for dynamic systems with non-additive process and measurement noise RVs is given by Algorithm 3.4. Note that no explicit calculations of Jacobian matrices are necessary for implementing this algorithm.

Consider the nonlinear dynamic system represented by Equations (3.22) and (3.23), which are rewritten here

$$\mathbf{x}\left(k\right) = \mathbf{f}\left(\mathbf{x}\left(k-1\right), \mathbf{u}\left(k-1\right), \vartheta\left(k-1\right)\right), \tag{3.63}$$

$$\mathbf{y}\left(k\right) = \mathbf{h}\left(\mathbf{x}\left(k\right), \xi\left(k\right)\right). \tag{3.64}$$

The state RV is redefined as the augmentation of the system state vector, and process and measurement noise vectors as follows,

$$\mathbf{x^a}(k) = \begin{bmatrix} \mathbf{x}(k) \\ \vartheta(k) \\ \xi(k) \end{bmatrix} \tag{3.65}$$

where the dimension of $\mathbf{x^a}(k)$ is given by $L = n + dim(\vartheta(k)) + dim(\xi(k))$. Furthermore, the augmented state covariance matrix is built from the covariance matrices of $\mathbf{x}(k)$, $\vartheta(k)$ and $\xi(k)$ as follows

$$\mathbf{P^a_x}(k) = \begin{bmatrix} \mathbf{P_x}(k) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R} \end{bmatrix}. \tag{3.66}$$

In the simplified case in which the process and measurement noise RVs are additive, the computational complexity of the UKF can be reduced [15, 19]. In such a case, the system state RV needs not to be augmented with the noise RVs. This reduces dimensionality of the sigma points as well as the total number of sigma points used. The covariances of the noise sources are then incorporated into the state covariance using a simple additive procedure [15]. The implementation of UKF for systems with additive process and measurement noise is given by Algorithm 3.5. The complexity of the algorithm is of order n^3. This is the same complexity as the EKF [15].

Algorithm 3.4 Unscented Kalman Filter (UKF) Algorithm for Systems with Non-Additive Noise.

- Initialization at $k = 0$

$$\hat{\mathbf{x}}(0) = E[\mathbf{x}(0)], \tag{3.67}$$

$$\mathbf{P}_{\mathbf{x}}(0) = E\left[(\mathbf{x}(0) - \hat{\mathbf{x}}(0))(\mathbf{x}(0) - \hat{\mathbf{x}}(0))^T\right], \tag{3.68}$$

$$\hat{\mathbf{x}}^{\mathbf{a}}(0) = E\left[\begin{array}{ccc} \hat{\mathbf{x}}^T(0) & \mathbf{0} & \mathbf{0} \end{array}\right]^T,$$

$$\mathbf{P}_{\mathbf{x}}^{\mathbf{a}}(0) = \left[\begin{array}{ccc} \mathbf{P}_{\mathbf{x}}(0) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R} \end{array}\right].$$

- For $k = 1, 2, \cdots, \infty$

1. Calculate $2L + 1$ sigma-points

$$\mathcal{X}_0(k-1) = \hat{\mathbf{x}}^{\mathbf{a}}(k-1), \tag{3.69}$$

$$\mathcal{X}_i(k-1) = \hat{\mathbf{x}}^{\mathbf{a}}(k-1) + \left(\sqrt{(L+\lambda)\mathbf{P}_{\mathbf{x}}^{\mathbf{a}}(k-1)}\right)_i, \quad i = 1, \cdots, L, \tag{3.70}$$

$$\mathcal{X}_i(k-1) = \hat{\mathbf{x}}^{\mathbf{a}}(k-1) - \left(\sqrt{(L+\lambda)\mathbf{P}_{\mathbf{x}}^{\mathbf{a}}(k-1)}\right)_{i-L}, \quad i = L+1, \cdots, 2L. \tag{3.71}$$

2. Time-update equations:

$$\mathcal{X}_i^{\mathbf{x}}(k) = \mathbf{f}\left(\mathcal{X}_i^{\mathbf{x}}(k-1), \mathbf{u}(k-1), \mathcal{X}_i^{\vartheta}(k-1)\right), \tag{3.72}$$

$$\hat{\mathbf{x}}^-(k) = \sum_{i=0}^{2L} \eta_i^m \mathcal{X}_i^{\mathbf{x}}(k), \tag{3.73}$$

$$\mathbf{P}_{\mathbf{x}}^-(k) = \sum_{i=0}^{2L} \eta_i^c \left(\mathcal{X}_i^{\mathbf{x}}(k) - \hat{\mathbf{x}}^-(k)\right)\left(\mathcal{X}_i^{\mathbf{x}}(k) - \hat{\mathbf{x}}^-(k)\right)^T. \tag{3.74}$$

3. Measurement-update equations:

$$\mathcal{Y}_i(k) = \mathbf{h}\left(\mathcal{X}_i^{\mathbf{x}}(k), \mathcal{X}_i^{\xi}(k)\right), \tag{3.75}$$

$$\hat{\mathbf{y}}^-(k) = \sum_{i=0}^{2L} \eta_i^m \mathcal{Y}_i(k), \tag{3.76}$$

$$\mathbf{P}_{\mathbf{y}}^-(k) = \sum_{i=0}^{2L} \eta_i^c \left(\mathcal{Y}_i(k) - \hat{\mathbf{y}}^-(k)\right)\left(\mathcal{Y}_i(k) - \hat{\mathbf{y}}^-(k)\right)^T, \tag{3.77}$$

$$\mathbf{P}_{\mathbf{xy}}^-(k) = \sum_{i=0}^{2L} \eta_i^c \left(\mathcal{X}_i^{\mathbf{x}}(k) - \hat{\mathbf{x}}^-(k)\right)\left(\mathcal{Y}_i(k) - \hat{\mathbf{y}}^-(k)\right)^T, \tag{3.78}$$

$$\mathbf{K}(k) = \mathbf{P}_{\mathbf{y}}^-(k)\mathbf{P}_{\mathbf{xy}}^-(k)^{-1}, \tag{3.79}$$

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}^-(k) + \mathbf{K}(k)\left(\mathbf{y}(k) - \hat{\mathbf{y}}^-(k)\right), \tag{3.80}$$

$$\mathbf{P}_{\mathbf{x}}(k) = \mathbf{P}_{\mathbf{x}}^-(k) - \mathbf{K}(k)\mathbf{P}_{\mathbf{y}}^-(k)\mathbf{K}(k)^T. \tag{3.81}$$

Algorithm 3.5 Unscented Kalman Filter (UKF) Algorithm for Systems with Additive Noise.

- Initialization at $k = 0$

$$\hat{\mathbf{x}}(0) = E[\mathbf{x}(0)], \tag{3.82}$$

$$\mathbf{P}_{\mathbf{x}}(0) = E\left[(\mathbf{x}(0) - \hat{\mathbf{x}}(0))(\mathbf{x}(0) - \hat{\mathbf{x}}(0))^T\right]. \tag{3.83}$$

- For $k = 1, 2, \cdots, \infty$

1. Calculate $2n + 1$ sigma-points

$$\mathcal{X}_0(k-1) = \hat{\mathbf{x}}(k-1), \tag{3.84}$$

$$\mathcal{X}_i(k-1) = \hat{\mathbf{x}}(k-1) + \left(\sqrt{(n+\lambda)\mathbf{P}_{\mathbf{x}}(k-1)}\right)_i, \quad i = 1, \cdots, n, \tag{3.85}$$

$$\mathcal{X}_i(k-1) = \hat{\mathbf{x}}(k-1) - \left(\sqrt{(n+\lambda)\mathbf{P}_{\mathbf{x}}(k-1)}\right)_{i-n}, \quad i = n+1, \cdots, 2n. \tag{3.86}$$

2. Time-update equations:

$$\mathcal{X}_i^*(k) = \mathbf{f}(\mathcal{X}_i(k-1), \mathbf{u}(k-1)), \tag{3.87}$$

$$\hat{\mathbf{x}}^-(k) = \sum_{i=0}^{2n} \eta_i^m \mathcal{X}_i^*(k), \tag{3.88}$$

$$\mathbf{P}_{\mathbf{x}}^-(k) = \mathbf{Q} + \sum_{i=0}^{2n} \eta_i^c \left(\mathcal{X}_i^*(k) - \hat{\mathbf{x}}^-(k)\right)\left(\mathcal{X}_i^*(k) - \hat{\mathbf{x}}^-(k)\right)^T. \tag{3.89}$$

3. Measurement-update equations:

$$\mathcal{X}_0(k) = \mathcal{X}_0^*, \tag{3.90}$$

$$\mathcal{X}_i(k) = \mathcal{X}_0^* + \left(\sqrt{(n+\lambda)\mathbf{P}_{\mathbf{x}}^-(k)}\right)_i, \quad i = 1, \cdots, n, \tag{3.91}$$

$$\mathcal{X}_i(k) = \mathcal{X}_0^* - \left(\sqrt{(n+\lambda)\mathbf{P}_{\mathbf{x}}^-(k)}\right)_{i-n}, \quad i = n+1, \cdots, 2n. \tag{3.92}$$

$$\mathcal{Y}_i(k) = \mathbf{h}(\mathcal{X}_i(k)), \tag{3.93}$$

$$\hat{\mathbf{y}}^-(k) = \sum_{i=0}^{2n} \eta_i^m \mathcal{Y}_i(k), \tag{3.94}$$

$$\mathbf{P}_{\mathbf{y}}^-(k) = \mathbf{R} + \sum_{i=0}^{2n} \eta_i^c \left(\mathcal{Y}_i(k) - \hat{\mathbf{y}}^-(k)\right)\left(\mathcal{Y}_i(k) - \hat{\mathbf{y}}^-(k)\right)^T, \tag{3.95}$$

$$\mathbf{P}_{\mathbf{xy}}^-(k) = \sum_{i=0}^{2n} \eta_i^c \left(\mathcal{X}_i(k) - \hat{\mathbf{x}}^-(k)\right)\left(\mathcal{Y}_i(k) - \hat{\mathbf{y}}^-(k)\right)^T, \tag{3.96}$$

$$\mathbf{K}(k) = \mathbf{P}_{\mathbf{y}}^-(k)\mathbf{P}_{xy}^-(k)^{-1}, \tag{3.97}$$

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}^-(k) + \mathbf{K}(k)\left(\mathbf{y}(k) - \hat{\mathbf{y}}^-(k)\right), \tag{3.98}$$

$$\mathbf{P}_{\mathbf{x}}(k) = \mathbf{P}_{\mathbf{x}}^-(k) - \mathbf{K}(k)\mathbf{P}_{\mathbf{y}}^-(k)\mathbf{K}(k)^T. \tag{3.99}$$

Chapter 4

Nonlinear Model Predictive Control

This chapter discusses the key concepts of *nonlinear model predictive control* (NMPC). Section 4.1 provides a short review of model predictive control theory. Section 4.2 is dedicated for the derivation of a prediction model of the NMPC. NMPC uses this model to determine predicted states/outputs which forms part of the information needed to construct the cost function. The cost function forms a critical part of the optimal control problem. It can be formulated differently for different problems, depending on the NMPC controller design specifications. The optimal control problem which NMPC solves in our case to determine the optimal sequence of control inputs is discussed in Section 4.4. The problem has to be converted into a standard nonlinear programming problem so that it can be solved by the available solution methods such as SQP. Section 4.5 provides the procedure of converting NMPC optimal control problem into the standard nonlinear programming problem. The last section is dedicated to providing the formulation of a constrained NMPC which systematically handles system constraints.

4.1 Introduction

The MPC also known as a *receding horizon predictive control* (RHPC) is categorized as the model-based control scheme. It employs an explicit model of the system as well as the measurements and the past sequence of control inputs to generate the predicted state trajectory of the system. The predicted trajectory is then used to compute the optimal sequence of control inputs to drive the system in the future. The prediction of state is performed over a finite future time interval known as a *prediction horizon*. A finite horizon optimal control problem which normally penalizes a *tracking error*, i.e., a distance of the predicted state trajectory from the desired state trajectory, and possibly the magnitudes of control inputs is formulated. Solving the optimal control problem while respecting system constraints yields an optimal sequence of control inputs. The first elements of the sequence are applied to the system as a feedback control in the next time step and the remaining elements are discarded. The horizon recedes one time step forward and the whole process is repeated.

Advantages that MPC has over other control schemes are: its ability to handle multi-variable interactions and constraints systematically in its framework [3, 5], systematic

handling of multivariate interactions, and use of future information when available. Constraints are present in almost all systems due to the physical and environmental limits on their operations. From economics point-of-view, profits are often maximized when the system operates on constraints boundaries [3]. There are many other cases in which performance improves as the system approaches its constraints, but the systems are seldom pushed to these constraints since damage may occur if any are violated [3]. MPC is ideal for such systems since it provides a systematic method for handling constraints.

The idea of this chapter is to show how NMPC problem is formulated. Many ideas that build up this chapter are from a book by Grune and Pannek [1] and an MIT masters thesis of Alaniz [3].

4.2 Prediction Modeling

NMPC problem can be separated into two main parts: a formulation of *prediction model* and a solution of the optimal control problem. The prediction model is formulated by iterating through the nominal model of the controlled system in order to predict the future state trajectory. A general form of discrete-time nonlinear nominal state space model is given by Equations (2.17) and (2.18) , which are repeated here,

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)), \tag{4.1}$$

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{x}(k)). \tag{4.2}$$

Notice that in this case, the model has been shifted on time step forward in order to make it suitable for prediction of the future states from the current ones. In contrast, the model given by Equations (2.17) and (2.18) is suitable for the prediction of the current states from the previous ones.

4.2.1 State Prediction Model

The future state prediction is achieved by iterating through the state transition equation given by Equation (4.1). The state prediction model is described as follows

$$\mathbf{x}(k+1|k) = \mathbf{f}(\mathbf{x}(k|k), \mathbf{u}(k|k)), \tag{4.3}$$

$$\mathbf{x}(k+2|k) = \mathbf{f}(\mathbf{x}(k+1|k), \mathbf{u}(k+1|k)), \tag{4.4}$$

$$\mathbf{x}(k+3|k) = \mathbf{f}(\mathbf{x}(k+2|k), \mathbf{u}(k+2|k)), \tag{4.5}$$

$$\vdots$$

$$\mathbf{x}(k+N_c|k) = \mathbf{f}(\mathbf{x}(k+N_c-1|k), \mathbf{u}(k+N_c-1|k)), \tag{4.6}$$

$$\vdots$$

$$\mathbf{x}(k+N_p|k) = \mathbf{f}(\mathbf{x}(k+N_p-1|k), \mathbf{u}(k+N_c-1|k)), \tag{4.7}$$

where "$|k$" means that the state is predicted at time instant k. This pattern is followed until a number of time steps in a prediction horizon N_p is reached. If $N_c < N_p$, the last set

of control values $\mathbf{u}(k+N_c-1|k)$ in the control sequence is maintained for the remaining $(N_p - N_c)$ time steps [3]. N_c is the control horizon. Note that in the formulation of NMPC, N_c is always set to be either less than or equal to N_p. This is shown by Equations (4.3) through (4.7). The predicted state trajectory of the system is a function of the current state $\mathbf{x}(k) = \mathbf{x}(k|k)$ and the predicted sequence of control inputs $(u(k|k), \cdots, u(k+N_c-1|k))$. The predicted state trajectory and sequence of control inputs can be put in vector forms as follows,

$$\mathbf{X}(k) = \begin{bmatrix} \mathbf{x}(k+1|k) \\ \mathbf{x}(k+2|k) \\ \mathbf{x}(k+3|k) \\ \mathbf{x}(k+4|k) \\ \vdots \\ \mathbf{x}(k+N_p|k) \end{bmatrix}, \tag{4.8}$$

$$\mathbf{U}(k) = \begin{bmatrix} \mathbf{u}(k|k) \\ \mathbf{u}(k+1|k) \\ \mathbf{u}(k+2|k) \\ \mathbf{u}(k+3|k) \\ \vdots \\ \mathbf{u}(k+N_p-1|k) \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{u}(k|k) \\ \mathbf{u}(k+1|k) \\ \vdots \\ \mathbf{u}(k+N_c-1|k) \end{bmatrix}. \tag{4.9}$$

Using Equations (4.8) and (4.9), Equations (4.3) through (4.7) can be summarized as the function of the current state $\mathbf{x}(k) = \mathbf{x}(k|k)$ and predicted control input vector $\mathbf{U}(k)$ as follows

$$\mathbf{X}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{U}(k)). \tag{4.10}$$

4.2.2 Output Prediction Model

The predicted state estimates given by Equation (4.8), but generated from Equation (4.10) are propagated through the output equation given by Equation (4.2) in order to determine the predicted output estimates of the system. An *output prediction model* is described as follows;

$$\mathbf{y}(k+1|k) = \mathbf{h}(\mathbf{x}(k+1|k)), \tag{4.11}$$
$$\mathbf{y}(k+2|k) = \mathbf{h}(\mathbf{x}(k+2|k)), \tag{4.12}$$
$$\mathbf{y}(k+3|k) = \mathbf{h}(\mathbf{x}(k+3|k)), \tag{4.13}$$
$$\mathbf{y}(k+4|k) = \mathbf{h}(\mathbf{x}(k+4|k)), \tag{4.14}$$
$$\vdots$$
$$\mathbf{y}(k+N_p|k) = \mathbf{h}(\mathbf{x}(k+N_p|k)). \tag{4.15}$$

In a similar fashion, this pattern is followed until the number of time steps in the prediction horizon is reached. The last set of output values in the predicted output trajectory is $\mathbf{y}(k+N_p|k)$. Notice that the predicted output trajectory is a function of the predicted

state trajectory. The predicted output estimates can be put in a vector form as follows

$$\mathbf{Y}(k) = \begin{bmatrix} \mathbf{y}(k+1|k) \\ \mathbf{y}(k+2|k) \\ \mathbf{y}(k+3|k) \\ \mathbf{y}(k+4|k) \\ \vdots \\ \mathbf{y}(k+N_p|k) \end{bmatrix}. \tag{4.16}$$

Using Equation (4.16), the output prediction model given by Equations (4.11) through (4.15) can be written in a compact form as follows

$$\mathbf{Y}(k) = \mathbf{h}(\mathbf{x}(k)). \tag{4.17}$$

NMPC algorithm basically solves an optimal control problem to determine an optimal sequence of control inputs such that the controlled system's future outputs track a given output trajectory which is given by

$$\mathbf{Y}^{sp}(k) = \begin{bmatrix} \mathbf{y}^{sp}(k+1|k) \\ \mathbf{y}^{sp}(k+2|k) \\ \mathbf{y}^{sp}(k+3|k) \\ \mathbf{y}^{sp}(k+4|k) \\ \vdots \\ \mathbf{y}^{sp}(k+N_p|k) \end{bmatrix}, \tag{4.18}$$

$\mathbf{Y}(k)$ and $\mathbf{U}(k)$ are the predicted output and control input trajectories that are generated by the NMPC algorithm at every time step k while $\mathbf{Y}^{sp}(k)$ is the desired output trajectory. These are fed into the NMPC algorithm at every time step in order to construct a corresponding *cost function* which forms part of optimal control problem. Subsection 4.3 details how the cost function is constructed.

4.3 Cost Function

The OCP that is discussed here is composed of a cost function taken from [3], but with some modifications on a term which penalizes the sequence of control inputs. The cost function is quadratic in nature and it is composed of two cost functions J_y and J_u. The cost function J_y penalizes a distance of the predicted output trajectory $\mathbf{Y}(k)$ from the desired output trajectory $\mathbf{Y}^{sp}(k)$. The quadratic form of J_y is weighted with positive (semi)-definite weighting matrices $(\mathbf{Q}(k), \cdots, \mathbf{Q}(k+N_p-1))$ and hence J_y is given by

$$J_y = \sum_{i=0}^{N_p-1} \left(\mathbf{y}(k+i|k) - \mathbf{y}^{sp}(k+i|k)\right)^T \mathbf{Q}(k+i) \left(\mathbf{y}(k+i|k) - \mathbf{y}^{sp}(k+i|k)\right). \tag{4.19}$$

When the matrices $(\mathbf{Q}(k), \cdots, \mathbf{Q}(k+N_p-1))$ are placed along the main diagonal of an appropriately sized positive (semi)-definite weighting matrix $\mathbf{Q_y}(k)$ as

$$\mathbf{Q_y}(k) = \begin{bmatrix} \mathbf{Q}(k) & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}(k+1) & \cdots & \vdots \\ \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{Q}(k+N_p-1) \end{bmatrix}, \tag{4.20}$$

the cost function J_y can be written in a compact form as follows

$$J_y = (\mathbf{Y}(k) - \mathbf{Y}^{sp}(k))^T \mathbf{Q_y}(k) (\mathbf{Y}(k) - \mathbf{Y}^{sp}(k)). \tag{4.21}$$

On the other hand, the cost function J_u penalizes magnitudes of control inputs in the control sequence $\mathbf{U}(k)$. The quadratic form of the cost function is weighted with the positive (semi)-definite weighting matrices $(\mathbf{R}(k), \cdots, \mathbf{R}(k+N_p-1))$ and hence J_u is given by

$$J_u = \sum_{i=0}^{N_p-1} \mathbf{u}(k+i|k)^T \mathbf{R}(k+i) \mathbf{u}(k+i|k). \tag{4.22}$$

When the matrices $(\mathbf{R}(k), \cdots, \mathbf{R}(k+N_p-1))$ are put along the main diagonal of the appropriately sized positive (semi)-definite weighting matrix $\mathbf{R_u}(k)$ as

$$\mathbf{R_u}(k) = \begin{bmatrix} \mathbf{R}(k) & 0 & \cdots & 0 \\ 0 & \mathbf{R}(k+1) & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & \mathbf{R}(k+N_p-1) \end{bmatrix}, \tag{4.23}$$

J_u can be rewritten in the compact form as follows

$$J_u = \mathbf{U}(k)^T \mathbf{R_u}(k) \mathbf{U}(k). \tag{4.24}$$

The total cost function J is therefore, given as a sum of J_y and J_u as follows

$$\begin{aligned} J &= J_y + J_u, \\ &= (\mathbf{Y}(k) - \mathbf{Y}^{sp}(k))^T \mathbf{Q_y}(k) (\mathbf{Y}(k) - \mathbf{Y}^{sp}(k)) + \mathbf{U}(k)^T \mathbf{R}_u(k) \mathbf{U}(k). \end{aligned} \tag{4.25}$$

The total cost function is always nonnegative and noncomplex, which means that it can either be zero or any positive real number. The predicted output trajectory $\mathbf{Y}(k)$ coincide exactly with the desired output trajectory $\mathbf{Y}^{sp}(k)$ if and only if $J_y = 0$. If $J_y > 0$, then it is known that $\mathbf{Y}(k)$ is away from $\mathbf{Y}^{sp}(k)$. These two situations can be explained mathematically using Equation (4.26) give below

$$J_y \begin{cases} = 0, & if \quad \mathbf{Y}(k) = \mathbf{Y}^{sp}(k), \\ > 0, & if \quad \mathbf{Y}(k) \neq \mathbf{Y}^{sp}(k). \end{cases} \tag{4.26}$$

4.4 Optimal Control Problem

A process of optimizing a performance of the controlled system corresponds to minimizing J with respect $\mathbf{U}(k)$ subjected to equality and inequality constraints of the system. A possible minimum value that J can attain is zero. This minimization problem is what is technically called an optimal control problem (OCP) for MPC.

The OCP can be formulated either as an *unconstrained* or *constrained* OCP, depending on whether the constrained satisfaction is desired or not. For the purpose of the work conducted in this thesis, the constrained OCP is used for the formulation of the constrained NMPC controller. The feature which makes NMPC schemes more attractive for the control of nonlinear dynamic systems is their abilities to handle system constraints in their formulations.

In the presence of constraints, the OCP problem is presented mathematically as follows,

$$\min_{\mathbf{U}(k)} \quad (\mathbf{Y}(k) - \mathbf{Y}^{sp}(k))^T \mathbf{Q}_{\mathbf{y}} (\mathbf{Y}(k) - \mathbf{Y}^{sp}(k)) + \mathbf{U}(k)^T \mathbf{R}_u \mathbf{U}(k), \tag{4.27}$$

subject to

$$\mathbf{x}(k+i|k) = \mathbf{f}(\mathbf{x}(k+i-1|k), \mathbf{u}(k+i-1|k)), \tag{4.28}$$

$$\mathbf{y}(k+i|k) = \mathbf{h}(\mathbf{x}(k+i-1|k)), \ i = 0, \cdots, N_p - 1, \tag{4.29}$$

$$\mathbf{u}(k+i|k) = \mathbf{u}(k+N_c-1|k), \ i = N_c, \cdots, N_p - 1, \tag{4.30}$$

$$\mathbf{X}_{min} \leq \mathbf{X}(k) \leq \mathbf{X}_{max}, \tag{4.31}$$

$$\mathbf{U}_{min} \leq \mathbf{U}(k) \leq \mathbf{U}_{max}, \tag{4.32}$$

$$\mathbf{Y}_{min} \leq \mathbf{Y}(k) \leq \mathbf{Y}_{max}, \tag{4.33}$$

where $\mathbf{X}_{min}$ and $\mathbf{X}_{max}$ are the lower and upper bounds of the state trajectory, respectively. $\mathbf{U}_{min}$ and $\mathbf{U}_{max}$ are the lower and upper bounds of the sequence of control inputs, respectively, and $\mathbf{Y}_{min}$ and $\mathbf{Y}_{max}$ are the lower and upper limits of the system output trajectory. Equations (4.31) through (4.33) describe the inequality constraints of the system. Equality constraints are defined by the system model.

4.5 Discretization of OCP

To solve OCP described by Equations (4.27) through (4.33) using solution methods, such as *sequential quadratic programming* (SQP) and *interior point method* (IMP), the OCP has to be transformed into a standard nonlinear programming problem (NLP). The process of transforming OCP into NLP is called *discretization* [1]. This process should not be confused with discretization of dynamic models discussed in Chapter 2. A standard NLP is described as follows,

$$\underset{\mathbf{z}(k)}{min} \quad \mathbf{F}(\mathbf{z}(k)), \tag{4.34}$$

subject to

$$\mathbf{H}(\mathbf{z}(k)) = 0, \tag{4.35}$$

$$\mathbf{G}(\mathbf{z}(k)) \leq 0, \tag{4.36}$$

where $\mathbf{F}$ is the cost function of the problem, and $\mathbf{H}$ and $\mathbf{G}$ are equality and inequality constraints, respectively. A variable $\mathbf{z}(k)$ is called an optimization variable. Either the cost function is nonlinear or one or more of the constraints are nonlinear so that the problem becomes nonlinear.

The sequence of control inputs $\mathbf{U}(k)$ that is regarded as the sequence of optimization variables in OCP is seen as an equivalence of $\mathbf{z}(k)$. The cost function J is seen as equivalence to $\mathbf{F}$ in NLP. The system dynamics constraints of OCP can be transformed into the equality constraints of the standard NLP by rewriting them as follows,

$$\mathbf{x}(k+i|k) - \mathbf{f}(\mathbf{x}(k+i-1|k), \mathbf{u}(k+i-1|k)) = \mathbf{0}, \tag{4.37}$$

$$\mathbf{y}(k+i|k) - \mathbf{h}(\mathbf{x}(k+i-1|k)) = \mathbf{0},\ i = 0, \cdots, N_p - 1, \tag{4.38}$$

$$\mathbf{u}(k+i|k) - \mathbf{u}(k+N_c-1|k) = \mathbf{0},\ i = N_c, \cdots, N_p - 1, \tag{4.39}$$

which can be put in a vector form in the following manner,

$$\mathbf{H} = \begin{bmatrix} \mathbf{x}(k+i|k) - \mathbf{f}(\mathbf{x}(k+i-1|k), \mathbf{u}(k+i-1|k)) \\ \mathbf{y}(k+i|k) - \mathbf{h}(\mathbf{x}(k+i-1|k)) \\ \mathbf{u}(k+i|k) - \mathbf{u}(k+N_c-1|k) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{4.40}$$

In a similar manner, inequality constraints of OCP can be transformed into the inequality constraints of the standard NLP by rewriting them as follows,

$$\begin{bmatrix} \mathbf{X}_L \\ -\mathbf{X}_U \end{bmatrix} + \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \mathbf{X}(k) \leq \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{4.41}$$

$$\begin{bmatrix} \mathbf{U}_L \\ -\mathbf{U}_U \end{bmatrix} + \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \mathbf{U}(k) \leq \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{4.42}$$

$$\begin{bmatrix} \mathbf{Y}_L \\ -\mathbf{Y}_U \end{bmatrix} + \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \mathbf{Y}(k) \leq \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{4.43}$$

which can also be concatenated in a vector form in the following manner

$$\mathbf{G} = \begin{bmatrix} \mathbf{X}_L \\ -\mathbf{X}_U \\ \mathbf{U}_L \\ -\mathbf{U}_U \\ \mathbf{Y}_L \\ -\mathbf{Y}_U \end{bmatrix} + \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{X}(k) \\ \mathbf{U}(k) \\ \mathbf{Y}(k) \end{bmatrix} \leq \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{4.44}$$

Equations (4.40) and (4.44) are now in a desired standard form similar to that of the constraints of NLP. Therefore the standard NLP is rewritten as,

$$\underset{\mathbf{U}(k)}{min} \; J(\mathbf{Y}(k), \mathbf{U}(k), k), \tag{4.45}$$

subject to

$$\begin{bmatrix} \mathbf{x}(k+i|k) - \mathbf{f}(\mathbf{x}(k+i-1|k), \mathbf{u}(k+i-1|k)) \\ \mathbf{y}(k+i|k) - \mathbf{h}(\mathbf{x}(k+i-1|k)) \\ \mathbf{u}(k+i|k) - \mathbf{u}(k+N_c-1|k) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{4.46}$$

$$\begin{bmatrix} \mathbf{X}_L \\ -\mathbf{X}_U \\ \mathbf{U}_L \\ -\mathbf{U}_U \\ \mathbf{Y}_L \\ -\mathbf{Y}_U \end{bmatrix} + \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{X}(k) \\ \mathbf{U}(k) \\ \mathbf{Y}(k) \end{bmatrix} \leq \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{4.47}$$

4.6 Constrained NMPC Algorithm

The general formulation of constrained NMPC algorithm is given by Algorithm 4.1. The assumptions made by the author about the controlled system are that:

- all state variables of the controlled system are observables,
- the controlled system is immune from the process and measurement noises,
- the controlled system is immune from the unknown disturbances,
- the model of the controlled system captures the dynamics of the system appropriately, i.e., no plant-model mismatch.

Algorithm 4.1 Constrained Nonlinear Model Predictive Control Algorithm.

– For $k = 1, 2, \cdots$:

1. Prediction model:

$$\mathbf{x}\left(k+i|k\right) = \mathbf{f}\left(\mathbf{x}\left(k+i-1|k\right), \mathbf{u}\left(k+i-1|k\right)\right), \tag{4.48}$$
$$\mathbf{y}\left(k+i|k\right) = \mathbf{h}\left(\mathbf{x}\left(k+i|k\right)\right), \qquad i = 1, \cdots N_p. \tag{4.49}$$

2. Solve the optimal control problem:

$$\underset{\mathbf{U}(k)}{min} \quad J\left(\mathbf{Y}\left(k\right), \mathbf{U}\left(k\right), k\right), \tag{4.50}$$

subject to

$$\begin{bmatrix} \mathbf{x}\left(k+i|k\right) - \mathbf{f}\left(\mathbf{x}\left(k+i-1|k\right), \mathbf{u}\left(k+i-1|k\right)\right) \\ \mathbf{y}(k+i) - \mathbf{h}\left(\mathbf{x}\left(k+i-1|k\right)\right) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{4.51}$$

$$\begin{bmatrix} \mathbf{X}_L \\ -\mathbf{X}_U \\ \mathbf{U}_L \\ -\mathbf{U}_U \\ \mathbf{Y}_L \\ -\mathbf{Y}_U \end{bmatrix} + \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{X}\left(k\right) \\ \mathbf{U}\left(k\right) \\ \mathbf{Y}\left(k\right) \end{bmatrix} \leq \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{4.52}$$

3. Define new NMPC feedback control values:

$$\mathbf{u}^*\left(k\right) = \mathbf{u}\left(0\right) \in \mathbf{U}\left(k\right). \tag{4.53}$$

4. Apply optimal control input to the system:

$$\mathbf{x}\left(k+1\right) = \mathbf{f}\left(\mathbf{x}\left(k\right), \mathbf{u}^*\left(k\right)\right). \tag{4.54}$$

5. Measure the current system outputs:

$$\mathbf{y}\left(k+1\right) = \mathbf{h}\left(\mathbf{x}\left(k+1\right)\right). \tag{4.55}$$

Figure 4.1 illustrates Algorithm 4.1 with a simple schematic diagram. The NMPC controller receives an output reference trajectory $\mathbf{Y}^{sp}$ of the controlled variables and the current state $\mathbf{x}(k)$ of the system. Using the current state, NMPC algorithm initializes a given open-loop finite horizon optimal control problem similar to the one given by Equations (4.50) through (4.52) to solve for the optimal sequence of control inputs $\mathbf{U}^*(k)$ which minimizes the Equation (4.50). The first elements of the sequence $\mathbf{u}(k|k)$ is implemented on the plant thus completing the closed-loop.

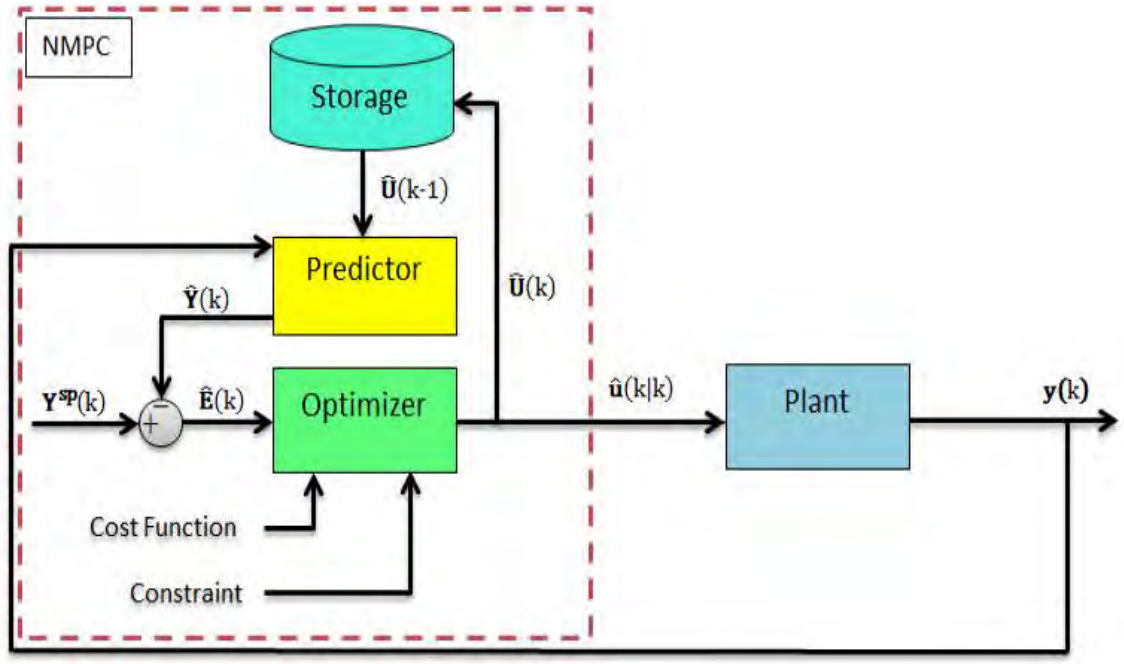


Figure 4.1: A block Diagram of the implementation configuration of NMPC.

Chapter 5

State Estimation Based NMPC Controller

In this thesis, the author proposes implementing EKF and UKF estimators for the design of state estimation based NMPC controller. Most importantly, the author has provided a literature review on how the standard EKF and UKF estimators are formulated in Chapter 3. Literature of NMPC controller was also provided in Chapter 4. In this chapter the author wants to integrate EKF and UKF into the formulation of NMPC to improve its robustness to unknown disturbances and plant-model mismatches. To accomplish that, some adjustments need to be done on the standard EKF and UKF algorithms. The algorithms are now formulated on the augmented state space models. The reason is so that the system states, unknown disturbances and unknown system parameters are estimated simultaneously. The author calls the resultant augmented EKF and UKF algorithms the Augmented State EKF (ASEKF) and Augmented State UKF (ASUKF) to distinguish them from the standard EKF and UKF, respectively.

Once the ASEKF and ASUKF estimators are ready, they are integrated into the formulation of NMPC controller. They are integrated such that they form part of the feedback loop of the NMPC. That is, the system measurements no longer go directly into the NMPC prediction model, but they have to be used by the estimator to jointly determine the estimates of unknown state variables, system parameters and disturbances. A vector containing these estimates is then sent to the prediction model. With these information the NMPC computes a control action to drive the controlled system to the reference behavior with no steady state offsets. Note that the prediction model is not formulated on the augmented state space model as do the ASEKF and ASUKF. Rather, no adjustments are done on it. The author calls the resultant state estimation based NMPC controllers the ASEKF-NMPC and ASUKF-NMPC controllers.

Section 5.1 presents designs of ASEKF and ASUKF estimators. Section 5.2 provides the case study which is aimed at validating the estimation performances of ASEKF and ASUKF. The section also provides the validation results. Section 5.3 presents the designs of ASEKF-NMPC and ASUKF-NMPC controllers. The controllers are validated for disturbance rejection and parametric plant-model mismatch handling performances in Section

5.4. The section also presents the validation results.

5.1 Designing Augmented State Estimation Methods

This section is divided into two subsections. The first subsection provides a design of ASEKF while the second one provides the design of ASUKF. Both ASEKF and ASUKF are formulated by adding some adjustments in Algorithms 3.2 and 3.5, respectively. The adjustments do not bring any complications into the standard algorithms. For instance, the augmented state space model described by Equations (2.32) and (2.33) simply replace Equations (2.17) and (2.18) in the algorithms. Furthermore, the state vector $\mathbf{x}(k)$ is replaced by an augmented state vector $\mathbf{x_a}(k)$ consisting of a vector of unknown system parameters $\mathbf{w}(k)$ and unknown output disturbance vector $\mathbf{d}(k)$, i.e.,

$$\mathbf{x_a}(k) = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{d}(k) \\ \mathbf{w}(k) \end{bmatrix}. \tag{5.1}$$

Notice that the input disturbance vector $\mathbf{v}(k)$ is not augmented in $\mathbf{x_a}(k)$ in this case. The reason is that augmenting for output disturbances already takes care of input disturbances of the system. Augmenting for input disturbances would therefore increase dimensionality of augmented state space model unnecessarily. The results of the validation will also be presented in Section 5.4.

Figure 5.1 illustrates the joint state, output disturbance and model parameter estimation using the proposed ASEKF and ASUKF state estimation methods. The state estimator receives the measurement $\mathbf{y}(k)$ from the plant and uses $\mathbf{y}(k)$ to determine a vector containing the estimates of the states, output disturbances and model parameters.

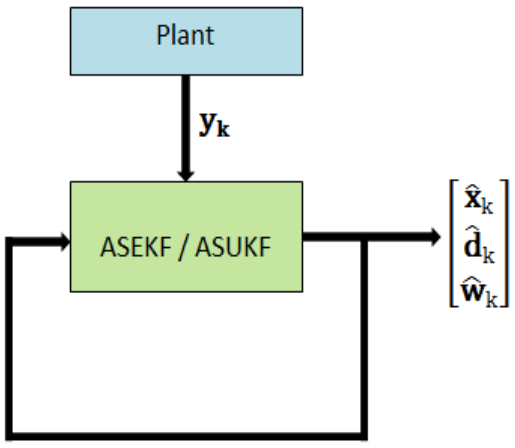


Figure 5.1: Joint State, Output Disturbance and Model Parameters Estimation.

The design objective is to extend the capabilities of the standard EKF and UKF estimators to estimating unknown disturbances and model parameters in addition to estimating unknown state variables from noisy measurements for any system.

5.1.1 Augmented State EKF (ASEKF)

ASEKF is found to be useful since it provides information about model mismatch and disturbances in addition to unknown state variables which can be provided by the standard EKF. ASEKF algorithm is derived directly from EKF algorithm by modifying some of its equations. Consider the augmented state space model described as follows,

$$\begin{bmatrix} \mathbf{x}(k) \\ \mathbf{d}(k) \\ \mathbf{w}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{f}(\mathbf{x}(k-1), \mathbf{u}(k-1), \mathbf{w}(k-1)) \\ \mathbf{d}(k-1) \\ \mathbf{w}(k-1) \end{bmatrix}, \tag{5.2}$$

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{x}(k), \mathbf{w}(k)) + \Gamma_{\mathbf{d}}\mathbf{d}(k), \tag{5.3}$$

which can be written compactly as,

$$x_{\mathbf{a}}(k) = \mathbf{f}_{\mathbf{a}}(\mathbf{x}_{\mathbf{a}}(k-1), \mathbf{u}(k-1)), \tag{5.4}$$

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{x}_{\mathbf{a}}(k)), \tag{5.5}$$

where $\mathbf{x_a}(k)$ is an augmented state vector and it is written as,

$$\mathbf{x_a}(k) = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{d}(k) \\ \mathbf{w}(k) \end{bmatrix}. \tag{5.6}$$

The functions $\mathbf{f_a}$ and $\mathbf{h_a}$ are augmented state transition and measurement functions, respectively, and they are assumed to be sufficiently differentiable and continuous in $\mathbf{x_a}$. Therefore, the process model and measurement model Jacobians are computed as follows,

$$\mathbf{J}_{\mathbf{f_x}} = \nabla_{\mathbf{x_a}}\mathbf{f_a}(\mathbf{x}, \mathbf{u}(k-1))|_{\mathbf{x}=\hat{\mathbf{x}}_{\mathbf{a}}(k-1)}, \tag{5.7}$$

$$\mathbf{J}_{\mathbf{h_x}} = \nabla_{\mathbf{x_a}}\mathbf{h}(\mathbf{x})|_{\mathbf{x}=\hat{\mathbf{x}}_{\mathbf{a}}^{-}(k)}. \tag{5.8}$$

With these Jacobians together with the augmented estimation error covariance and augmented process noise covariance matrices given as follows,

$$\mathbf{P_a}(k-1) = \begin{bmatrix} \mathbf{P_x}(k-1) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P_d}(k-1) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P_w}(k-1) \end{bmatrix}, \quad \mathbf{Q_a} = \begin{bmatrix} \mathbf{Q_x} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q_d} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q_w} \end{bmatrix}, \tag{5.9}$$

Algorithm 3.2 is modified and the result is Algorithm 5.1. The matrices $\mathbf{P_x}(k-1)$, $\mathbf{P_d}(k-1)$ and $\mathbf{P_w}(k-1)$ are *a priori* estimation error covariances for the state, output disturbance and model parameter estimates, respectively. The matrices, $\mathbf{Q_x}$, $\mathbf{Q_d}$ and $\mathbf{Q_w}$ are process noise covariances corresponding to the state, output disturbance and model parameters, respectively, and they are assumed to be constant. Based on Equations (5.7) through (5.9), the Kalman filter equations are applied recursively as in Algorithm 3.2.

Algorithm 5.1 Augmented State EKF (ASEKF) for System with Additive Noise.

- Initialization at $k = 0$

$$\hat{\mathbf{x}}(0) = E[\mathbf{x}(0)], \qquad \hat{\mathbf{d}}(0) = E[\mathbf{d}(0)], \qquad \hat{\mathbf{w}}(0) = E[\mathbf{w}(0)], \tag{5.10}$$

$$\hat{\mathbf{x}}_{\mathbf{a}}(0) = \begin{bmatrix} \hat{\mathbf{x}}(0)^T & \hat{\mathbf{d}}(0)^T & \hat{\mathbf{w}}(0)^T \end{bmatrix}^T, \tag{5.11}$$

$$\mathbf{P}_{\mathbf{x}}(0) = E\left[(\mathbf{x}(0) - \hat{\mathbf{x}}(0))(\mathbf{x}(0) - \hat{\mathbf{x}}(0))^T\right], \tag{5.12}$$

$$\mathbf{P}_{\mathbf{d}}(0) = E\left[\left(\mathbf{d}(0) - \hat{\mathbf{d}}(0)\right)\left(\mathbf{d}(0) - \hat{\mathbf{d}}(0)\right)^T\right], \tag{5.13}$$

$$\mathbf{P}_{\mathbf{w}}(0) = E\left[(\mathbf{w}(0) - \hat{\mathbf{w}}(0))(\mathbf{w}(0) - \hat{\mathbf{w}}(0))^T\right], \tag{5.14}$$

$$\mathbf{P}_{\mathbf{a}}(0) = \begin{bmatrix} \mathbf{P}_{\mathbf{x}}(0) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{\mathbf{d}}(0) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_{\mathbf{w}}(0) \end{bmatrix}, \qquad \mathbf{Q}_{\mathbf{a}} = \begin{bmatrix} \mathbf{Q}_{\mathbf{x}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{\mathbf{d}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_{\mathbf{w}} \end{bmatrix}. \tag{5.15}$$

- For $k = 1, \cdots, \infty$:

  1. Time-update equations:

$$\mathbf{J}_{\mathbf{f}_{\mathbf{x}}} = \nabla_{\mathbf{x}_{\mathbf{a}}} \mathbf{f}_{\mathbf{a}}(\mathbf{x}, \mathbf{u}(k-1))\,|_{\mathbf{x}=\hat{\mathbf{x}}_{\mathbf{a}}(k-1)}, \tag{5.16}$$

$$\hat{\mathbf{x}}_{\mathbf{a}}^{-}(k) = \mathbf{f}_{\mathbf{a}}(\hat{\mathbf{x}}_{\mathbf{a}}(k-1), \mathbf{u}(k-1)), \tag{5.17}$$

$$\mathbf{P}_{\mathbf{a}}^{-}(k) = \mathbf{J}_{\mathbf{f}_{\mathbf{x}}} \mathbf{P}_{\mathbf{a}}(k-1) \mathbf{J}_{\mathbf{f}_{\mathbf{x}}}^T + \mathbf{Q}_{\mathbf{a}}. \tag{5.18}$$

  2. Measurement-update equations:

$$\mathbf{J}_{\mathbf{h}_{\mathbf{x}}} = \nabla_{\mathbf{x}} \mathbf{h}(\mathbf{x})\,|_{\mathbf{x}=\hat{\mathbf{x}}_{\mathbf{a}}^{-}(k)}, \tag{5.19}$$

$$\mathbf{K} = \mathbf{P}_{\mathbf{a}}^{-}(k) \mathbf{J}_{\mathbf{h}_{\mathbf{x}}}^T \left[\mathbf{J}_{\mathbf{h}_{\mathbf{x}}} \mathbf{P}_{\mathbf{a}}^{-}(k) \mathbf{J}_{\mathbf{h}_{\mathbf{x}}}^T + \mathbf{R}\right]^{-1}, \tag{5.20}$$

$$\hat{\mathbf{y}}^{-}(k) = \mathbf{h}\left(\hat{\mathbf{x}}_{\mathbf{a}}^{-}(k)\right), \tag{5.21}$$

$$\hat{\mathbf{x}}_{\mathbf{a}}(k) = \hat{\mathbf{x}}_{\mathbf{a}}^{-}(k) + \mathbf{K}(k)\left[\mathbf{y}(k) - \hat{\mathbf{y}}^{-}(k)\right], \tag{5.22}$$

$$\mathbf{P}_{\mathbf{a}}(k) = \left[\mathbf{I} - \mathbf{K}(k+1)\mathbf{J}_{\mathbf{h}_{\mathbf{x}}}\right] \mathbf{P}_{\mathbf{a}}^{-}(k). \tag{5.23}$$

Algorithm 5.1 is implemented on Matlab and then validated for output disturbance rejection and plant-model mismatch performance on the Van der Pol oscillator in Section 5.2. From the validation results, the author can establish whether the proposed estimation algorithms are suitable for being implemented in the design of state estimation-based NMPC controller in Section 5.3.

5.1.2 Augmented State UKF (ASUKF)

Similarly, ASUKF is useful since it provides information about disturbances and model mismatch in addition to the estimates of unknown state variables. ASUKF is derived from the standard UKF algorithm and it is accomplished through replacing some equations of Algorithm 3.5. Intuitively, the reader can expect that ASUKF algorithm would provide better estimates compared to ASEKF algorithm since it is based on the UKF algorithm which is known to have been developed to address the major short coming of EKF algorithm [18].

Consider the augmented state space model described by Equations (5.4) and (5.5), and $\mathbf{x}_{\mathbf{a}}(k)$, $\mathbf{P}_{\mathbf{a}}(k)$ and $\mathbf{Q}_{\mathbf{a}}$ given by Equations (5.6) and (5.9), respectively. Then, the $2L + 1$

sigma points $\mathcal{X}_i$ and the weighting vectors η^m and η^c are now computed as,

$$\mathcal{X}_0 = \bar{\mathbf{x}}_{\mathbf{a}}(k), \quad i = 0, \tag{5.24}$$

$$\mathcal{X}_i = \bar{\mathbf{x}}_{\mathbf{a}}(k) + \left(\sqrt{(L+\lambda)\mathbf{P}_{\mathbf{a}}(k)}\right)_i, \quad i = 1, \cdots, L, \tag{5.25}$$

$$\mathcal{X}_i = \bar{\mathbf{x}}_{\mathbf{a}}(k) - \left(\sqrt{(L+\lambda)\mathbf{P}_{\mathbf{a}}(k)}\right)_i, \quad i = L+1, \cdots, 2L, \tag{5.26}$$

$$\eta_0^m = \frac{\lambda}{\lambda + L}, \tag{5.27}$$

$$\eta_0^c = \frac{\lambda}{\lambda + L} + 1 - \alpha^2 + \beta, \tag{5.28}$$

$$\eta_i^m = \eta_i^c = \frac{1}{2[L+\lambda]}, \; i = 1, \cdots, 2L, \tag{5.29}$$

where L is the length of $\mathbf{x}_{\mathbf{a}}(k)$. Algorithm 3.5 can now be modified into Algorithm 5.2.

Algorithm 5.2 Augmented State UKF (ASUKF) for Systems with Additive Noise.

- Initialization at $k = 0$:

$$\hat{\mathbf{x}}(0) = E[\mathbf{x}(0)], \qquad \hat{\mathbf{d}}(0) = E[\mathbf{d}(0)], \qquad \hat{\mathbf{w}}(0) = E[\mathbf{w}(0)], \tag{5.30}$$

$$\hat{\mathbf{x}}_{\mathbf{a}}(0) = \begin{bmatrix} \hat{\mathbf{x}}(0)^T & \hat{\mathbf{d}}(0)^T & \hat{\mathbf{w}}(0)^T \end{bmatrix}^T, \tag{5.31}$$

$$\mathbf{P}_{\mathbf{x}}(0) = E\left[(\mathbf{x}(0) - \hat{\mathbf{x}}(0))(\mathbf{x}(0) - \hat{\mathbf{x}}(0))^T\right], \tag{5.32}$$

$$\mathbf{P}_{\mathbf{d}}(0) = E\left[\left(\mathbf{d}(0) - \hat{\mathbf{d}}(0)\right)\left(\mathbf{d}(0) - \hat{\mathbf{d}}(0)\right)^T\right], \tag{5.33}$$

$$\mathbf{P}_{\mathbf{w}}(0) = E\left[(\mathbf{w}(0) - \hat{\mathbf{w}}(0))(\mathbf{w}(0) - \hat{\mathbf{w}}(0))^T\right], \tag{5.34}$$

$$\mathbf{P}_{\mathbf{a}}(0) = \begin{bmatrix} \mathbf{P}_{\mathbf{x}}(0) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{\mathbf{d}}(0) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{P}_{\mathbf{w}}(0) \end{bmatrix}, \qquad \mathbf{Q}_{\mathbf{a}} = \begin{bmatrix} \mathbf{Q}_{\mathbf{x}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_{\mathbf{d}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Q}_{\mathbf{w}} \end{bmatrix}. \tag{5.35}$$

- For $k = 1, 2, \cdots, \infty$

1. Calculate $2L + 1$ sigma-points:

$$\mathcal{X}_0(k-1) = \hat{\mathbf{x}}_{\mathbf{a}}(k-1), \tag{5.36}$$

$$\mathcal{X}_i(k-1) = \hat{\mathbf{x}}_{\mathbf{a}}(k-1) + \left(\sqrt{(L+\lambda)\mathbf{P}_{\mathbf{a}}(k-1)}\right)_i, \quad i = 1, \cdots, L, \tag{5.37}$$

$$\mathcal{X}_i(k-1) = \hat{\mathbf{x}}_{\mathbf{a}}(k-1) - \left(\sqrt{(L+\lambda)\mathbf{P}_{\mathbf{a}}(k-1)}\right)_{i-L}, \quad i = L+1, \cdots, 2L. \tag{5.38}$$

2. Time-update equations:

$$\mathcal{X}_i^*(k) = \mathbf{f}_{\mathbf{a}}(\mathcal{X}_i(k-1), \mathbf{u}(k-1)), \tag{5.39}$$

$$\hat{\mathbf{x}}_{\mathbf{a}}^-(k) = \sum_{i=0}^{2n} \eta_i^m \mathcal{X}_i^*(k), \tag{5.40}$$

$$\mathbf{P}_{\mathbf{a}}^-(k) = \mathbf{Q}_{\mathbf{a}} + \sum_{i=0}^{2n} \eta_i^c \left(\mathcal{X}_i^*(k) - \hat{\mathbf{x}}_{\mathbf{a}}^-(k)\right)\left(\mathcal{X}_i^*(k) - \hat{\mathbf{x}}_{\mathbf{a}}^-(k)\right)^T. \tag{5.41}$$

3. Measurement-update equations:

$$\mathcal{X}_0(k) = \mathcal{X}_0^*(k), \tag{5.42}$$

$$\mathcal{X}_i(k) = \mathcal{X}_0^*(k) + \left(\sqrt{(L+\lambda)\mathbf{P}_{\mathbf{a}}^-(k)}\right)_i, \quad i = 1, \cdots, L, \tag{5.43}$$

$$\mathcal{X}_i(k) = \mathcal{X}_0^*(k) - \left(\sqrt{(L+\lambda)\mathbf{P}_{\mathbf{a}}^-(k)}\right)_{i-L}, \quad i = L+1, \cdots, 2L. \tag{5.44}$$

$$\mathcal{Y}_i(k) = \mathbf{h}(\mathcal{X}_i(k)), \tag{5.45}$$

$$\hat{\mathbf{y}}^-(k) = \sum_{i=0}^{2L} \eta_i^m \mathcal{Y}_i(k), \tag{5.46}$$

$$\mathbf{P}_{\mathbf{a},\mathbf{y}}^-(k) = \mathbf{R} + \sum_{i=0}^{2L} \eta_i^c \left(\mathcal{Y}_i(k) - \hat{\mathbf{y}}_{\mathbf{a}}^-(k)\right)\left(\mathcal{Y}_i(k) - \hat{\mathbf{y}}_{\mathbf{a}}^-(k)\right)^T, \tag{5.47}$$

$$\mathbf{P}_{\mathbf{a},\mathbf{xy}}^-(k) = \sum_{i=0}^{2n} \eta_i^c \left(\mathcal{X}_i(k) - \hat{\mathbf{x}}_{\mathbf{a}}^-(k)\right)\left(\mathcal{Y}_i(k) - \hat{\mathbf{y}}_{\mathbf{a}}^-(k)\right)^T, \tag{5.48}$$

$$\mathbf{K}(k) = \mathbf{P}_{\mathbf{a},\mathbf{y}}^-(k)\,\mathbf{P}_{\mathbf{a},\mathbf{xy}}(k)^{-1}, \tag{5.49}$$

$$\hat{\mathbf{x}}_{\mathbf{a}}(k) = \hat{\mathbf{x}}_{\mathbf{a}}^-(k) + \mathbf{K}(k)\left(\mathbf{y}(k) - \hat{\mathbf{y}}^-(k)\right), \tag{5.50}$$

$$\mathbf{P}_{\mathbf{a}}(k) = \mathbf{P}_{\mathbf{a}}^-(k) - \mathbf{K}(k)\,\mathbf{P}_{\mathbf{a},\mathbf{y}}(k)\,\mathbf{K}(k)^T. \tag{5.51}$$

5.2 Case Study: Van der Pol Oscillator

Van der Pol oscillator is used as one of the example problems in [16] to study the behavior of UKF and to compare robustness of UKF with that of EKF to modeling errors. In this case study, the author uses Van der Pol oscillator to validated the performances of ASEKF and ASUKF estimators developed above for disturbance rejection and plant-model mismatch handling. Van der Pol oscillator is a highly nonlinear system and therefore it is sufficient to perform the validations of the nonlinear estimators on it. The dynamics of the oscillator are described by the following state space equations

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ \mu\left(1 - x_1^2(t)\right) x_2(t) - x_1(t) \end{bmatrix}, \tag{5.52}$$

$$y(t) = x_1(t), \tag{5.53}$$

where $x_1(t)$ and $x_2(t)$ are state variables and $\mu > 0$ is a sole parameter of the system. Equation (5.53) is the output equation. Notice that it has been assumed that only $x_1(t)$ is an observable state while $x_2(t)$ is not.

The system is then discretized using backward Euler method at the sampling time interval of $T = 0.05\,sec$. The resultant state space model is then augmented for the output step disturbance and parameter models to form the augmented state space model,

$$\begin{bmatrix} x_1(k) \\ x_2(k) \\ d(k) \\ \mu(k) \end{bmatrix} = \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \\ d(k-1) \\ \mu(k-1) \end{bmatrix} + 0.05 \begin{bmatrix} x_2(k-1) \\ \mu\left(1 - x_1^2(k-1)\right) x_2(k-1) - x_1(k-1) \\ 0 \\ 0 \end{bmatrix}, \tag{5.54}$$

$$y(k) = x_1(k) + d(k). \tag{5.55}$$

The augmented state space model is used in ASEKF and ASUKF to obtain the state estimates, $\hat{x}_1(k)$ and $\hat{x}_2(k)$, output disturbance estimate $\hat{d}(k)$ and parameter estimate $\hat{\mu}(k)$. The simulation time is set to be $30\,sec$ throughout. The augmented estimation error covariance, augmented process noise covariance and measurement noise covariance matrices were determined through trial and error method which is often not accurate. The tuning parameter values of the ASEKF and ASUKF algorithms are given in Table 5.1.

The performance validations of the estimators have been studied under different conditions (i.e. large initial state estimate deviation, output step disturbance and parametric plant-model mismatch) with the time responses of the state, disturbance and parameter estimates. Their relative performances in output disturbance rejection in the presence of parametric plant-model mismatch has been studied with mean square error (MSE) used as the performance index. The simulation results with respect to the ASEKF and ASUKF estimators are presented graphically in the following subsections.

Table 5.1: Tuning Parameters used in ASEKF and ASUKF algorithm.

Augmented noise covariance matrices used in ASEKF and ASUKF

1. *Disturbance rejection in the absence of model mismatch*

$\mathbf{Q_a} = diag\left\{ \begin{array}{cccc} 2\times10^{-5} & 2\times10^{-5} & 2\times10^{-5} \end{array} \right\}$

$\mathbf{R} = 0.001$

2. *Disturbance rejection in the presence of model mismatch*

$\mathbf{Q_a} = diag\left\{ \begin{array}{cccc} 2\times10^{-5} & 2\times10^{-5} & 2\times10^{-5} & 2\times10^{-5} \end{array} \right\}$

$\mathbf{R} = 0.001$

Initial augmented state error covariance matrix used in ASEKF and ASUKF

1. *Disturbance rejection in the absence of model mismatch*

$\mathbf{P_a} = diag\left\{ \begin{array}{ccc} 0.1 & 0.1 & 0.1 \end{array} \right\}$

2. *Disturbance rejection in the presence of model mismatch*

$\mathbf{P_a} = diag\left\{ \begin{array}{cccc} 0.1 & 0.1 & 0.1 & 0.1 \end{array} \right\}$

Addition tuning parameters for ASUKF algorithm

$\alpha = 0.01 \qquad \beta = 2 \qquad \kappa = 0$

Initial state and initial augmented state estimate

$\mathbf{x}(0) = \left[\begin{array}{cc} 0.5 & -1 \end{array} \right]^T$

1. *Disturbance rejection in the absence of model mismatch*

$\hat{\mathbf{x}}_\mathbf{a}(0) = \left[\begin{array}{ccc} 0.5 & 1 & 0 \end{array} \right]^T$

2. *Disturbance rejection in the presence of model mismatch*

$\hat{\mathbf{x}}_\mathbf{a}(0) = \left[\begin{array}{cccc} 0.5 & 1 & 0 & 0 \end{array} \right]^T$

5.2.1 Simulation Results with ASEKF

Firstly, the ASEKF algorithm is validated under the conditions of large initial estimate deviations and output step disturbance. The aim is to determine if the estimates will eventually converge to their true values from large initial estimate deviations. In this case, no model mismatches are present. The results of the simulation are provided by Figures 5.2 and 5.3. Figure 5.2 shows the time responses of the true states of Van der Pol oscillator and their estimates generated using ASEKF. Even though the estimates deviate significantly at the beginning of the simulation as the ASEKF and Van der Pol oscillator have different initial conditions, it can be seen that they converge to their respective true states as time progresses. This indicates that ASEKF indeed minimizes the estimation error for both $x_1(k)$ and $x_2(k)$.

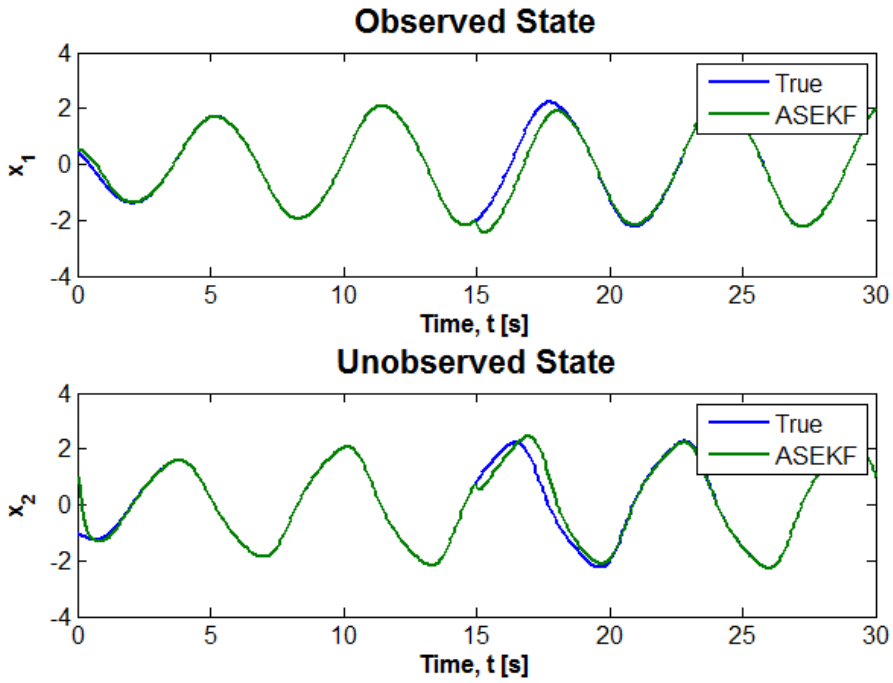


Figure 5.2: ASEKF state estimates with large initial estimate deviations.

At 15 *sec* of the simulation, the output step disturbance of magnitude of 2 is injected into the system. It can be seen from Figure 5.3 that ASEKF is able to satisfactorily estimate output step disturbance. An effect of good disturbance estimation can be seen in Figure 5.2. When the disturbance is introduced, both state estimates quickly drift away from their respective true states, but return close to them again with time. Another important observation is that initially the disturbance estimate is zero as does initial true output disturbance. See Figure 5.3. The estimate quickly drifts ways from the true disturbance for sometime and then returns close to zero. This can be attributed to the fact that since ASEKF performs state and disturbance estimation simultaneously, perturbations in state estimates influence perturbations in disturbance estimate.

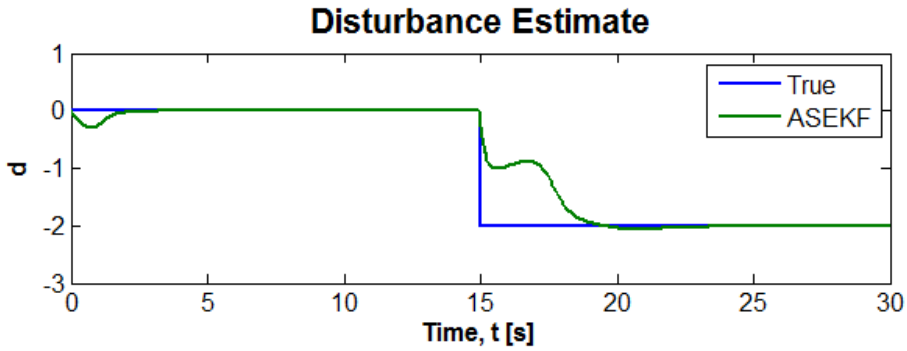


Figure 5.3: ASEKF output disturbance estimate with large initial estimate deviations.

To make ASEKF estimation problem even more interesting, a 5% plant-model mismatch is introduced to the model parameter, μ. The simulations are run again under similar conditions. Figures 5.4 through 5.6 present results of the simulation. From Figure 5.4, it can be observed that both state estimates converge to their respective true states in the presence of 5% model mismatch. The speed of convergence is slightly reduced. This

observation is obvious when comparing the plots in Figure 5.2 relative to the plots in Figure 5.4. This is attributed to the fact that it takes ASEKF sometime before it obtains the best estimates for μ and d simultaneously. Another reason is that since both μ and d are augmented to the same state vector, dimensionality of the model increases and so does the computational time of ASEKF.

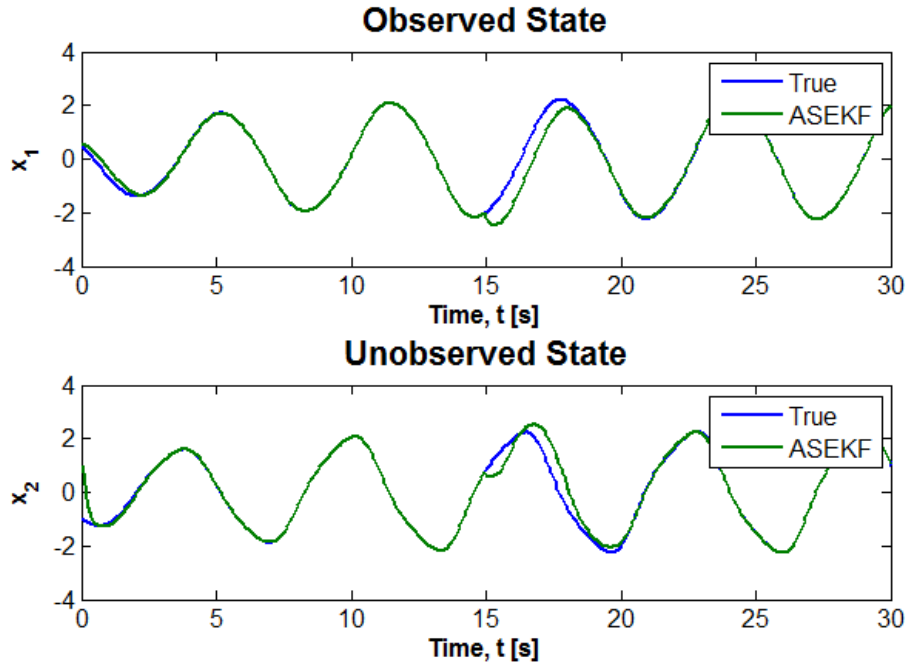


Figure 5.4: ASEKF state estimates with large initial estimate deviations and 5% model mismatch.

This is an interesting behavior because it shows that when a joint state, disturbance and parameter estimation is performed, the rate of convergence of the estimates reduces. Figure 5.5 depicts time responses of the true output disturbance and its ASEKF estimate. From these figure it can be concluded that ASEKF achieved good disturbance estimation. In general, ASEKF is able to reject output disturbances occurring on the system in the presence of model mismatch. Figure 5.6 illustrates an estimation of parameter μ using ASEKF. The large state estimation deviations at the beginning of the simulation and just after the step disturbance is introduced can be observed. Those are influenced by the fact that μ is treated as an additional state of the system to be estimated.

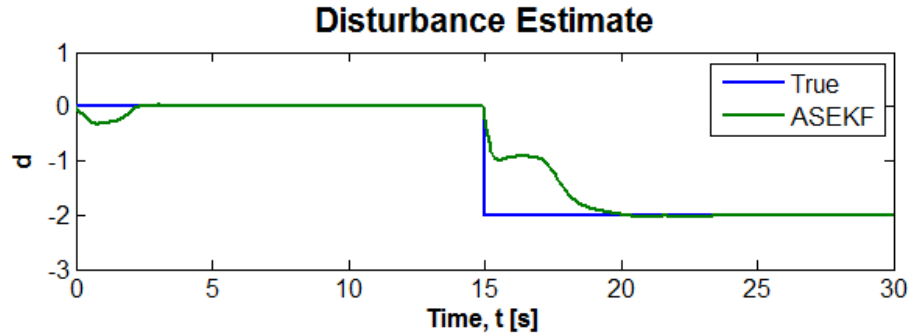


Figure 5.5: ASEKF disturbance estimate with large initial estimate deviations and 5% model mismatch.

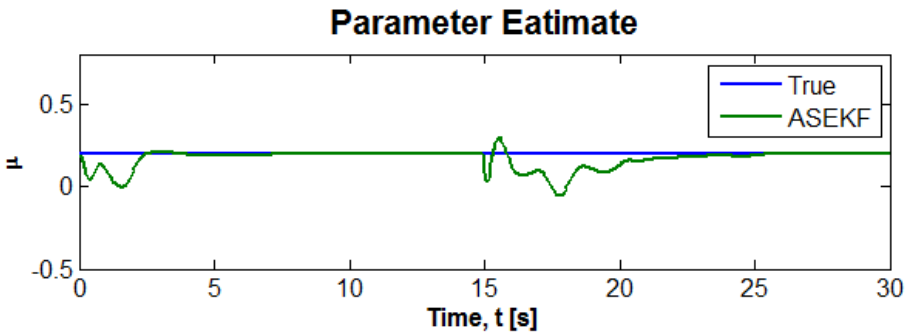


Figure 5.6: ASEKF estimate with large initial estimate deviations and 5% model mismatch.

5.2.2 Simulation Results with ASUKF

Figures 5.7 and 5.8 depict validation results of ASUKF under the conditions of large initial estimate deviations and output step disturbance. Looking at Figure 5.7, ASUKF is able to accommodate large initial estimate deviations since the state estimates satisfactorily converge closer to the true states as expected.

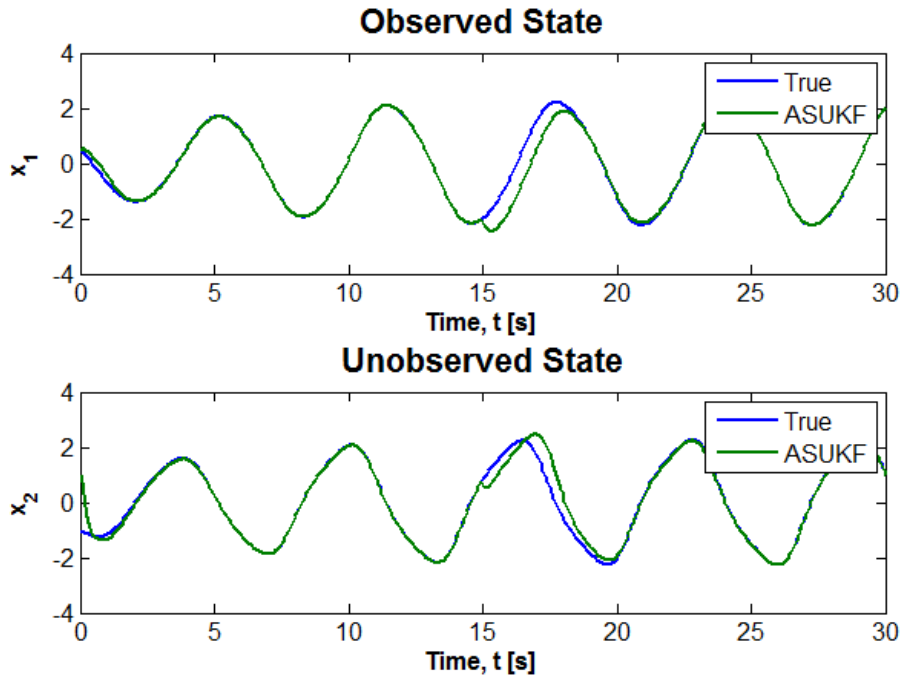


Figure 5.7: ASUKF state estimates with large initial estimation deviations.

The output step disturbance of magnitude 2 is injected into the Van der Pol oscillator at 15 *sec* time instant. From Figure 5.8, it is evident that ASUKF is able to force the output disturbance estimate towards the true output disturbance, thereby letting the system state estimates converge to their respective true states. In general, the results of ASUKF are similar to those of ASEKF in Figures 5.2 and 5.3. Therefore it can be concluded that under the conditions of large state estimate deviations and output step disturbance of magnitude of 2, both filters are able to achieve similar disturbance rejection performance.

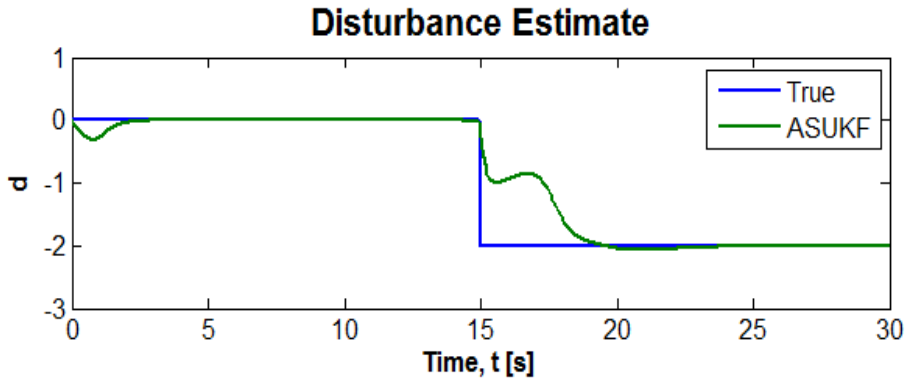


Figure 5.8: ASUKF disturbance estimate with large initial estimation deviations.

To assess the performance of ASUKF for disturbance rejection in the presence of model mismatch, a 5% parametric plant-model mismatch was introduced. It can be seen from Figure 5.9 shows that ASUKF is able to satisfactorily estimate the states in the presence of 5% model mismatch. Similar to the case of ASEKF, the rate of convergence of the state estimates to the true values has slightly reduced in comparison to the previous case. The true output step disturbance and its estimate are displayed in Figure 5.10. The figure shows that ASUKF estimates output step disturbance well in the presence of 5% model mismatch.

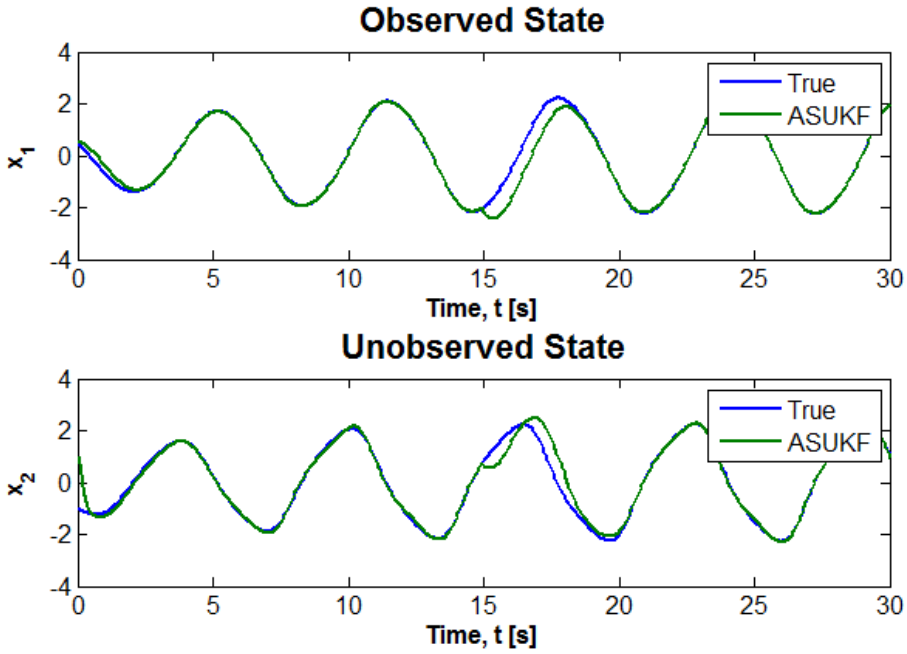


Figure 5.9: ASUKF state estimates with large initial estimation deviations and model mismatch.

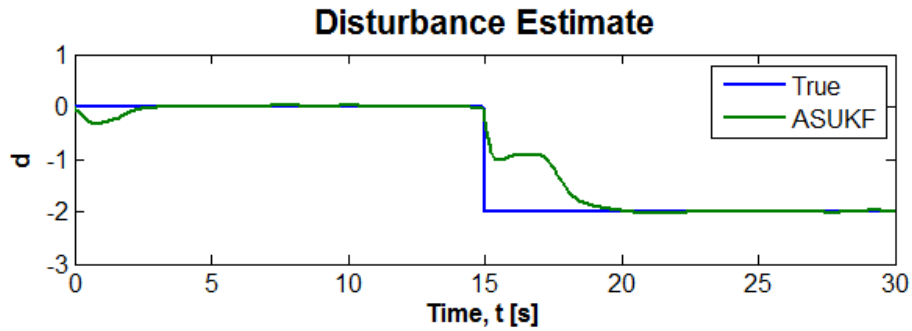


Figure 5.10: ASUKF disturbance estimate with large initial estimation deviations and 5% model mismatch.

The estimation of μ is illustrated in Figure 5.11. From the figure, it can be seen that ASUKF estimator was not able to provide good estimate for μ. That could have resulted from the bad tuning of the filter since trial and error approach was used. ASUKF has additional parameter, α, β and κ, which need to be specified appropriately to achieve its optimal performance. That did not affect the estimation of system states and output disturbance as it has been seen in Figures 5.9 and 5.10.

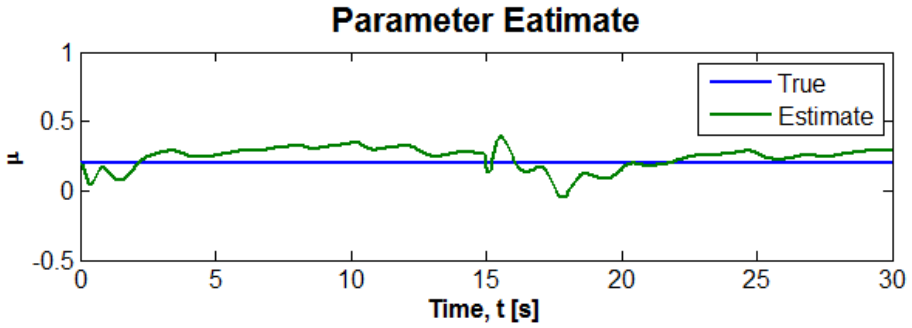


Figure 5.11: ASUKF parameter estimate with large initial estimate deviations and 5% model mismatch.

The relative performance of the ASUKF and ASEKF in output disturbance rejection in the presence of 5% model mismatch was conducted. Figure 5.12 depicts the MSE's of the state estimates and it shows that in general, ASEKF gives the minimum MSE compared to ASUKF. Possibly, this is the result of ASUKF's bad joint parameter estimation.

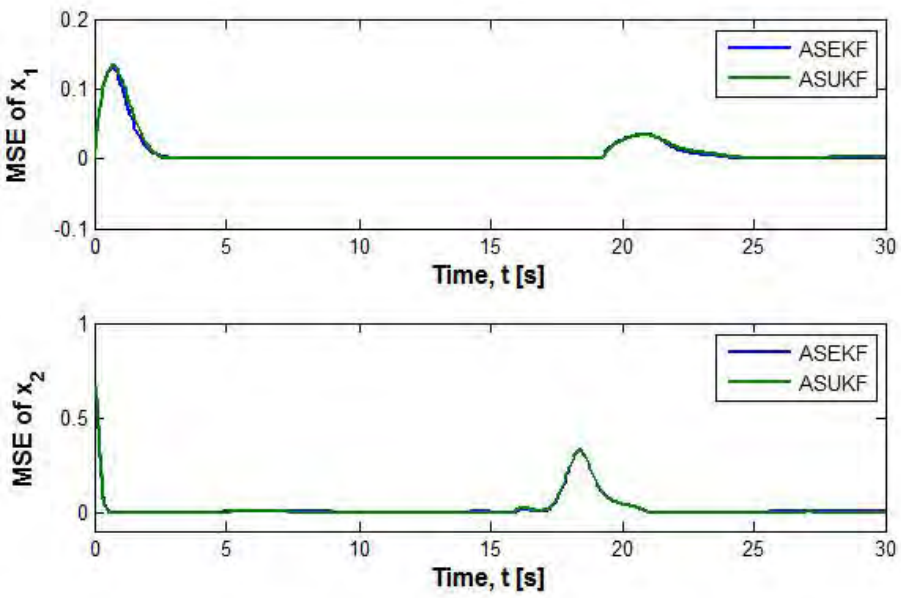


Figure 5.12: Comparison of ASEKF and ASUKF performances in output disturbance in the presence of 5% model mismatch.

5.3 Designing State Estimation Based NMPC Controllers

ASEKF and ASUKF estimators developed in the previous section are employed for the design of state estimation based NMPC controllers in this section. The resultant controllers are called ASEKF-NMPC and ASUKF-NMPC controllers, depending on the state estimator used in its formulation. They are formulated by modifying a feedback loop of the NMPC algorithm given by Algorithm 4.1 by adding ASEKF and ASUKF, respectively. The system measurements are now used by the estimator to determine the estimates of unknown state variables, output disturbances and model parameters. New formulations can guarantee the closed-loop stability and robustness of NMPC to output disturbances and parametric plant-model mismatches. ASEKF-NMPC and ASUKF-NMPC controllers are therefore more practical than the NMPC controller discussed in Chapter 4.

Figure 5.13 shows the schematic diagram of the proposed state estimation based NMPC controller, i.e., ASEKF-NMPC controller or ASUKF-NMPC controller. The figure is intended to help explain how the two controllers are formulated. The output disturbance $\mathbf{d}(k)$ comes into the control system as an additive term to the plant measurements to form $\mathbf{y}(k)$. $\mathbf{y}(k)$ is then fed into the state estimator, ASEKF / ASUKF block, to generate a vector containing the estimates of unknown state variables, unknown output disturbances and unknown model parameters. The estimation information is then send to the NMPC prediction model, *Predictor block,* to compute the predicted output trajectory $\hat{\mathbf{Y}}_{\mathbf{a}}(k)$ which together with the past sequence of control inputs $\hat{\mathbf{U}}(k-1)$ and output reference trajectory $\mathbf{Y}^{sp}(k)$, are then utilized by the *Optimizer block* to compute new optimal sequence of control inputs $\hat{\mathbf{U}}(k)$. The *Optimizer block* is equipped with user defined cost function and system constraints which together form NMPC optimal control problem.

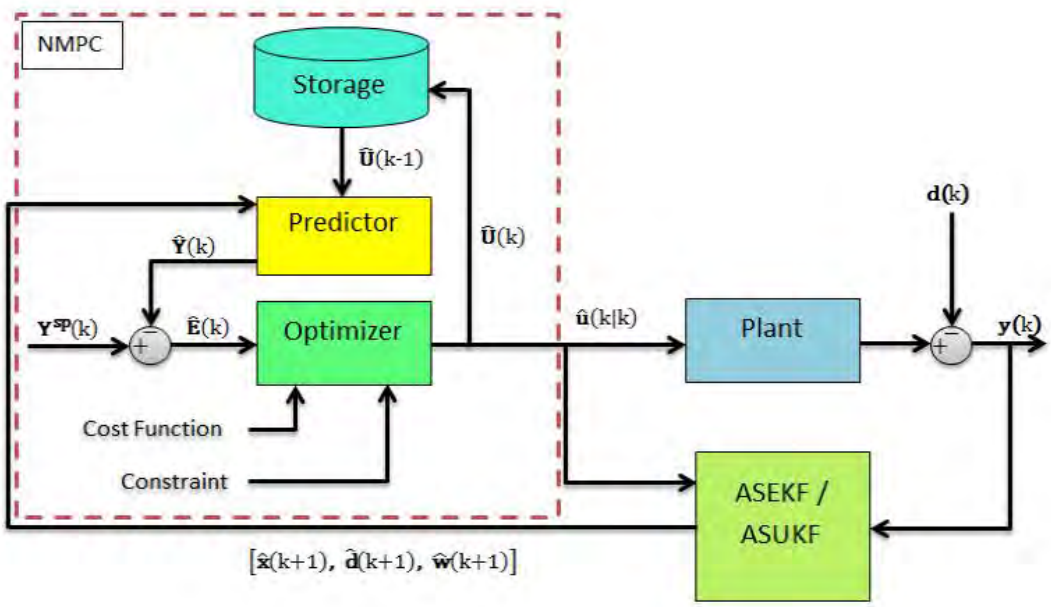


Figure 5.13: A Block Diagram of the Implementation Configuration of State Estimation Based NMPC Controller.

5.3.1 Prediction Model

The state prediction model of the state estimation based NMPC controller is formulated as follows,

$$\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k+1|k) = \mathbf{f}\left(\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k|k), \mathbf{u}(k|k), \hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{w}}(k)\right), \tag{5.56}$$

$$\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k+2|k) = \mathbf{f}\left(\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k+1|k), \mathbf{u}(k+1|k), \hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{w}}(k)\right), \tag{5.57}$$

$$\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k+3|k) = \mathbf{f}\left(\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k+1|k), \mathbf{u}(k+1|k), \hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{w}}(k)\right), \tag{5.58}$$

$$\vdots$$

$$\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k+N_c|k) = \mathbf{f}\left(\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k+N_c-1|k), \mathbf{u}(k+N_c-1|k), \hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{w}}(k)\right), \tag{5.59}$$

$$\vdots$$

$$\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k+N_p|k) = \mathbf{f}\left(\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k+N_p-1|k), \mathbf{u}(k+N_c-1|k), \hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{w}}(k)\right), \tag{5.60}$$

while the output prediction model is formulated as follows,

$$\hat{\mathbf{y}}(k+1|k) = \mathbf{h}\left(\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k+1|k)\right) + \Gamma_{\mathbf{d}}\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{d}}(k), \tag{5.61}$$

$$\hat{\mathbf{y}}(k+2|k) = \mathbf{h}\left(\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k+2|k)\right) + \Gamma_{\mathbf{d}}\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{d}}(k), \tag{5.62}$$

$$\hat{\mathbf{y}}(k+3|k) = \mathbf{h}\left(\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k+3|k)\right) + \Gamma_{\mathbf{d}}\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{d}}(k), \tag{5.63}$$

$$\hat{\mathbf{y}}(k+4|k) = \mathbf{h}\left(\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k+4|k)\right) + \Gamma_{\mathbf{d}}\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{d}}(k), \tag{5.64}$$

$$\vdots$$

$$\hat{\mathbf{y}}(k+N_p|k) = \mathbf{h}\left(\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k+N_p|k)\right) + \Gamma_{\mathbf{d}}\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{d}}(k). \tag{5.65}$$

Notice that in the prediction model, the output disturbance and model parameter estimates, $\hat{\mathbf{d}}(k) = \hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{d}}(k)$ and $\hat{\mathbf{w}}(k) = \hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{w}}(k)$, respectively, do not evolve over N_p since they are not the actual states of the system. The predicted state estimate trajectory and the

resultant predicted output trajectory can be put in vector forms, $\hat{\mathbf{X}}(k)$ and $\hat{\mathbf{Y}}(k)$, respectively.

5.3.2 Cost Function

The total cost function is the sum of the expected value of the weighted 2-norm of the tracking error and the weighted 2-norm of the magnitudes of control inputs,

$$E\left[J\right] = E\left[\left(\hat{\mathbf{Y}}\left(k\right) - \mathbf{Y}^{sp}\left(k\right)\right)\mathbf{Q_y}\left(\hat{\mathbf{Y}}\left(k\right) - \mathbf{Y}^{sp}\left(k\right)\right)^{T}\right] + \mathbf{U}\left(k\right)\mathbf{R}_u\mathbf{U}\left(k\right)^{T}. \tag{5.66}$$

5.3.3 Optimal Control Problem

From Equation (5.66), the optimal control problem is formulated as a constrained problem as in Section (4.5) using the system dynamic model as equality constraints and variable bounds as inequality constraints. The resultant optimal control problem given by Equations (5.67) through (5.69) as follows

$$\underset{\mathbf{U}(k)}{min}\quad E\left[\left(\hat{\mathbf{Y}}\left(k\right) - \mathbf{Y}^{sp}\left(k\right)\right)\mathbf{Q_y}\left(\hat{\mathbf{Y}}\left(k\right) - \mathbf{Y}^{sp}\left(k\right)\right)^{T}\right] + \mathbf{U}\left(k\right)\mathbf{R}_u\mathbf{U}\left(k\right)^{T}, \tag{5.67}$$

subject to

$$\begin{bmatrix} \hat{\mathbf{x}}^{\mathbf{x}}_{\mathbf{a}}\left(k+i|k\right) - \mathbf{f}\left(\hat{\mathbf{x}}^{\mathbf{x}}_{\mathbf{a}}\left(k+i-1|k\right), \mathbf{u}\left(k+i-1|k\right), \hat{\mathbf{x}}^{\mathbf{d}}_{\mathbf{a}}\left(k\right)\right) \\ \hat{\mathbf{y}}\left(k+i|k\right) - \mathbf{h}\left(\hat{\mathbf{x}}^{\mathbf{x}}_{\mathbf{a}}\left(k+i|k\right)\right) + \Gamma_{\mathbf{d}}\hat{\mathbf{x}}^{\mathbf{d}}_{\mathbf{a}}\left(k\right), \ i = 0, \cdots, N_p - 1 \\ \mathbf{u}\left(k+i|k\right) = \mathbf{u}(k+N_c-1|k), \ i = N_c, \cdots, N_p - 1, \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{5.68}$$

$$\begin{bmatrix} \mathbf{X}_L \\ -\mathbf{X}_U \\ \mathbf{U}_L \\ -\mathbf{U}_U \\ \mathbf{Y}_L \\ -\mathbf{Y}_U \end{bmatrix} + \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{X}}\left(k\right) \\ \mathbf{U}\left(k\right) \\ \hat{\mathbf{Y}}\left(k\right) \end{bmatrix} \leq \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{5.69}$$

5.3.4 State Estimation-based NMPC Algorithm

The state estimation based NMPC controllers (ASEKF-NMPC and ASUKF-NMPC) extends capabilities of the standard NMPC controller to rejecting unknown disturbances and handling plant-model mismatches which always exist when modeling dynamic systems. The implementable formulation of the proposed ASEKF-NMPC and ASUKF-NMPC controllers are represented by Algorithm 5.3.

Algorithm 5.3 Constrained ASEKF-NMPC algorithm.

- For $k = 1, 2, \cdots$:

1. Obtain augmented state estimate:

$$\hat{\mathbf{x}}_{\mathbf{a}}(k) = \begin{bmatrix} \hat{\mathbf{x}}(k)^T & \hat{\mathbf{d}}(k)^T & \hat{\mathbf{w}}(k)^T \end{bmatrix}^T. \tag{5.70}$$

2. Prediction model:

$$\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k+i|k) = \mathbf{f}\left(\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k+i-1|k), \mathbf{u}(k+i-1|k), \hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{w}}(k)\right), \tag{5.71}$$

$$\hat{\mathbf{y}}(k+i|k) = \mathbf{h}\left(\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k+i|k)\right) + \Gamma_{\mathbf{d}}\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{d}}(k), \quad i = 1, \cdots, N_p. \tag{5.72}$$

3. Solve the optimal control problem:

$$\min_{\mathbf{U}(k)} \quad E\left[\left(\hat{\mathbf{Y}}(k) - \mathbf{Y}^{sp}(k)\right)\mathbf{Q}_y\left(\hat{\mathbf{Y}}(k) - \mathbf{Y}^{sp}(k)\right)^T\right] + \mathbf{U}(k)\mathbf{R}_u\mathbf{U}(k)^T \tag{5.73}$$

subject to

$$\begin{bmatrix} \hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k+i|k) - \mathbf{f}\left(\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k+i-1|k), \mathbf{u}(k+i-1|k), \hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{d}}(k)\right) \\ \hat{\mathbf{y}}(k+i|k) - \mathbf{h}\left(\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{x}}(k+i|k)\right) + \Gamma_{\mathbf{d}}\hat{\mathbf{x}}_{\mathbf{a}}^{\mathbf{d}}(k) \\ \mathbf{u}(k+i|k) = \mathbf{u}(k+N_c-1|k) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \tag{5.74}$$

$$\begin{bmatrix} \hat{\mathbf{X}}_L \\ -\hat{\mathbf{X}}_U \\ \mathbf{U}_L \\ -\mathbf{U}_U \\ \hat{\mathbf{Y}}_L \\ -\hat{\mathbf{Y}}_U \end{bmatrix} + \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{X}}(k) \\ \mathbf{U}(k) \\ \hat{\mathbf{Y}}(k) \end{bmatrix} \leq \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{5.75}$$

4. Define a new NMPC feedback control values:

$$\mathbf{u}^*(k) = \mathbf{u}(0) \in \mathbf{U}(k). \tag{5.76}$$

5. Apply optimal control input to the system:

$$\mathbf{x}(k+1) = \mathbf{f}\left(\mathbf{x}(k), \mathbf{u}^*(k)\right) + \vartheta(k). \tag{5.77}$$

6. Measure the outputs:

$$\mathbf{y}(k+1) = \mathbf{h}\left(\mathbf{x}(k+1)\right) + \mathbf{d}(k+1) + \xi(k+1). \tag{5.78}$$

5.4 Case Study: Magnetic Levitation System

The closed-loop performances of ASEKF-NMPC and ASUKF-NMPC controllers are validated by applying them for the control of magnetic levitation system. The derivation of the model of magnetic levitation system is give below.

5.4.1 Modeling Magnetic Levitation System

Magnetic levitation system consists of electromagnet suspended above a position sensor which is mounted on a base of the system. In between electromagnet and sensor there

is a finite gap called an air gap within which a ferromagnetic ball moves up and down under influence of electromagnetic force. The electromagnetic force is generated by electromagnet and therefore, the electromagnet is seen as an actuator. The position sensor determines position of ferromagnetic ball in the air gap. To fix ferromagnetic ball in one position in the air gap, current flowing through the electromagnet is adjusted such that electromagnetic force balances weight of the ball and thus, the ball levitates in equilibrium position. Magnetic levitation system is a nonlinear, open-loop unstable system that need good dynamic model and controller [26]. Figure 5.14 illustrates the schematic diagram of magnetic levitation as viewed in a plane.

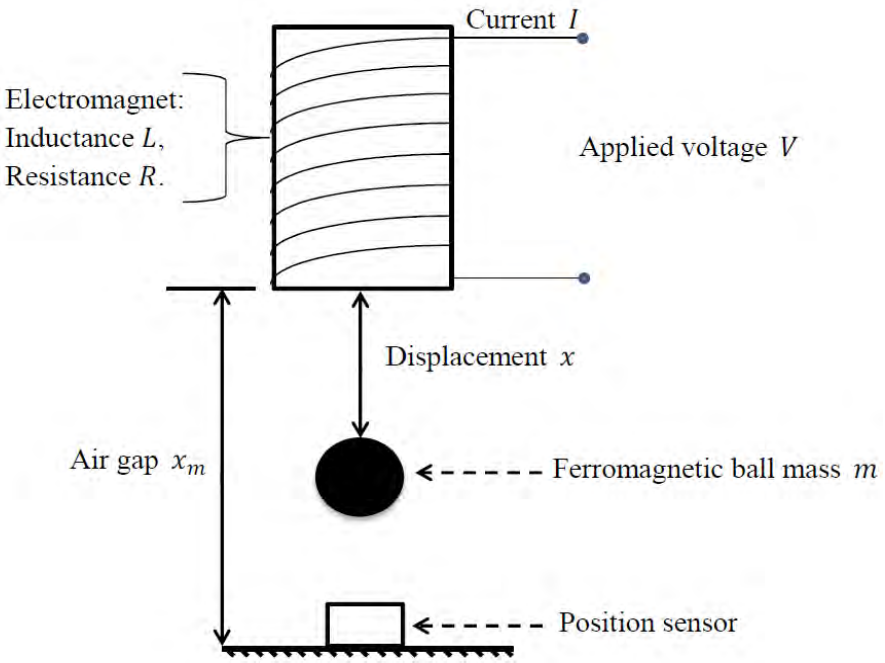


Figure 5.14: Schematic diagram of magnetic levitation system.

To control magnetic levitation system using nonlinear control methods, its nonlinear dynamic equations have to be derived. The nonlinear dynamics are derived by applying Newton's laws of motion for mechanical subsystem and Kirchhoff's voltage law for electromagnetic subsystem. Free body diagrams for mechanical subsystem and electromagnetic subsystem are shown in Figure 5.15 and they are used in derivation of nonlinear model.

The forces acting vertically on ferromagnetic ball are summed together according to Newton's second law of motion. A force balance equation given by Equation (5.79) is a result of the summation of forces. A difference $x_m - x(t)$ represents the displacement of the ball in the air-gap measured from the lower surface of the electromagnet. On the other hand, a flow of current in the electrical subsystem is governed by voltage balance equation given by Equation (5.80). Equation (5.80) is derived such that it obeys Kirchhoff's voltage law.

$$\sum F_y(t) = mg - c\frac{d}{dt}\left[x_m - x(t)\right] - F_{em}\left(x(t),\, I(t)\right) = m\frac{d^2}{dt^2}\left[x_m - x(t)\right], \tag{5.79}$$

$$\sum V_v(t) = RI(t) + \frac{d}{dt}\left[L\left(x(t)\right) I(t)\right] = V(t), \tag{5.80}$$

where m is the mass of the ferromagnetic ball, g is the gravitational constant, F_{em} is an

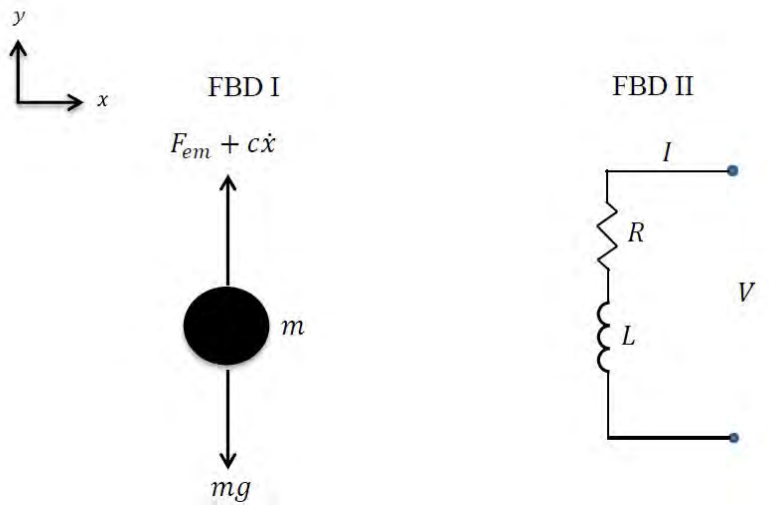


Figure 5.15: The free body diagrams of mechanical and electromagnetic components of magnetic levitation system.

electromagnetic force acting on the ball and it is expressed as a function of coil current $I(t)$ and a distance of the ball from the surface of electromagnet, $x(t)$. An expression for electromagnetic force is given by Equation (5.81) where C is a magnetic force constant. A total inductance L of the system is expressed as a nonlinear function of $x(t)$ and it can be approximated linearly as sum of coil inductance, L_c, (which is constant) and additional inductance influenced by the presence of the ball in magnetic field. Equation (5.82) gives total inductance.

$$F_{em}\left(x(t),\, I(t)\right) = C\left[\frac{I(t)}{x_m - x(t)}\right]^2, \tag{5.81}$$

$$L\left(x(t)\right) \approx L_c + \frac{2C}{x_m - x(t)}. \tag{5.82}$$

Substituting Equations (5.81) and (5.82) into Equations (5.79) and (5.80), respectively, results in

$$mg - c\frac{d}{dt}\left[x_m - x(t)\right] - C\left[\frac{I(t)}{x_m - x(t)}\right]^2 = m\frac{d^2}{dt^2}\left[x_m - x(t)\right], \tag{5.83}$$

$$RI(t) + \frac{d}{dt}\left[L_c\dot{I}(t) - 2C\frac{I(t)}{x_m - x\left(t\right)}\right] = V(t). \tag{5.84}$$

Equations (5.83) and (5.84) can be simplified and rearranged by solving for $\ddot{x}(t)$ and $\dot{I}(t)$. Making $\ddot{x}(t)$ and $\dot{I}(t)$ subjects of the formula in Equations (5.83) and (5.84) results in

$$\ddot{x}(t) = -g + \frac{C}{m}\left[\frac{I(t)}{x_m - x(t)}\right]^2 - \frac{c\dot{x}(t)}{m}, \tag{5.85}$$

$$\dot{I}(t) = -\frac{\left[R\left(x_m - x(t)\right)^2 + 2C\dot{x}\left(t\right)\right]}{\left(x_m - x\left(t\right)\right)\left(L_c\left(x_m - x\left(t\right)\right) + 2C\right)}I(t) + \left[\frac{x_m - x\left(t\right)}{L_c\left(x_m - x\left(t\right)\right) + 2C}\right]V(t). \tag{5.86}$$

The state space model of magnetic levitation system is constructed from Equations (5.85)-(5.86) by introducing internal variables shown in Equation (5.87) and replacing applied

voltage $V(t)$ by control variable $u(t)$.

$$x_1(t) = x(t), \quad x_2(t) = \dot{x}(t), \quad x_3(t) = I(t). \tag{5.87}$$

Equation (5.88) is the result of substituting internal variables into Equations (5.85) and (5.86) and this is called state space representation of magnetic levitation system

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ -g + \frac{C}{m}\left(\frac{x_3(t)}{x_m - x_1(t)}\right)^2 + \frac{cx_2(t)}{m} \\ -\frac{\left[R(x_m - x_1(t))^2 + 2Cx_2(t)\right]}{(x_m - x_1(t))(L_c(x_m - x_1(t)) + 2C)} x_3(t) + \left[\frac{x_m - x_1(t)}{L_c(x_m - x_1(t)) + 2C}\right] u(t) \end{bmatrix}, \tag{5.88}$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}. \tag{5.89}$$

The Euler method is applied to the model described by Equations (5.88) and (5.89) at the sampling time of $T = 0.01\,sec$ to find its discrete time counterpart. The resultant discrete time model is used to design the prediction model of the NMPC controller. Furthermore, the discrete time model is augmented for unknown output disturbance and unknown model parameters and then used to design ASEKF and ASUKF subsystems of ASEKF-NMPC and ASUKF-NMPC controllers, respectively. The constraints are applied based on the input saturation and the maximum allowable distance of a ferromagnetic ball within the air gap. The input constraint limits the amount of voltage supplied to the electromagnet between $[0,\ 10]\,V$. In the similar manner, the output constraint limit the displacement of the ferromagnetic ball between $[0,\ 0.12]\,m$.

The simulation parameters of magnetic levitation system considered here are listed in Table 5.2.

Table 5.2: Simulation parameters of magnetic levitation system.

Parameters	**Units**	**Values**
m	kg	0.02
g	m/s^2	9.81
R	Ω	10
L	H	0.01
C	N/A^2	0.005
c	N	0.01
x_m	m	0.12

5.4.2 NMPC Controller Setup

The main objective of the controller in this case is to lift a ferromagnetic ball vertically from a position close to the ground and let it track a given reference behavior between the ground and lower surface of electromagnet. The closed-loop performances are validated through a series of simulations under the conditions of noise, unknown output disturbance and plant-model mismatch. The purpose of validating under these conditions is to demonstrate the capabilities of ASEKF-NMPC and ASUKF-NMPC controllers. Unless otherwise stated,

unknown output disturbance in this validation is a step function of unknown magnitude. The noise is assumed to be zero-mean Gaussian white and it is distributed over the system outputs with standard deviation of 0.2% of the nominal value. Furthermore, the plant-model mismatch is a parametric plant-model mismatch. That is, the uncertainty in the parameters of the magnetic levitation system is considered. The tuning parameters of the ASEKF-NMPC and ASUKF-NMPC controllers are tuned through trial and error to obtain satisfactorily closed-loop setpoint tracking and disturbance rejection.

Reference Trajectory

A reference trajectory that the ferromagnetic ball is supposed to track is given by Equation (5.90). All state variables in reference trajectory are zero except for position and coil current. The coil current reference trajectory is given by $(x_m - x_{ref}(k))\sqrt{mg/C}$ where x_m is a maximum air gap and $x_{ref}(k)$ is a reference position. When the ball tracks the reference position trajectory accurately, its velocity $x_2(k)$ reduced to zero so that it stabilized there. At the same time, the coil current maintains a particular value until a perturbation occurs in the ball's position

$$\mathbf{X}^{sp} = \begin{bmatrix} x_{ref} & x_{ref} & \cdots & x_{ref} \\ 0 & 0 & \cdots & 0 \\ (x_m - x_{ref})\sqrt{mg/C} & (x_m - x_{ref})\sqrt{mg/C} & \cdots & (x_m - x_{ref})\sqrt{mg/C} \end{bmatrix}. \quad (5.90)$$

Controller Parameters

The tuning parameter of the ASEKF-NMPC and ASUKF-NMPC controllers are tuned accordingly through trial and error to get the satisfactory closed-loop performances of the controllers. The tuning parameters are the two weighting matrices, $\mathbf{Q}$ and $\mathbf{R}$ of the cost function, N_p, N_c, three covariance matrices of ASEKF and ASUKF estimators, $\mathbf{P_a}$, $\mathbf{Q_a}$ and $\mathbf{R_a}$, and additional parameters of ASUKF, α, β and κ. The cost function weighting matrices are tuned once and remain constant throughout the simulations. Table 5.3 summaries the values taken by the tuning parameters to conduct the simulations under different conditions. The table also provides the information about the initial state estimate $\mathbf{x}(0)$ and initial augmented state estimate $\hat{\mathbf{x}}_\mathbf{a}(0)$ used under different conditions. The sampling time used is $T = 0.01\,sec$ and it is common throughout all validation simulation runs.

Table 5.3: Tuning Parameters used in ASEKF-NMPC and ASUKF-NMPC controllers.

Tuning parameters used in NMPC part of the controllers

$\mathbf{Q} = diag\left\{ \begin{array}{ccc} 2.5 \times 10^5 & 1 & 1 \end{array} \right\}$

$\mathbf{R} = 0.001$

$N_p = 15$

$N_c = 7$

Tuning parameters used in ASEKF and ASUKF parts of the controllers

1. *Non-augmented system model*

$\mathbf{P_a} = diag\left\{ \begin{array}{ccc} 0.1 & 0.1 & 0.1 \end{array} \right\}$

$\mathbf{Q_a} = diag\left\{ \begin{array}{ccc} 0.002 & 0.002 & 0.002 \end{array} \right\}$

$\mathbf{R_a} = diag\left\{ \begin{array}{cc} 0.001 & 0.001 \end{array} \right\}$

2. *Model augmented for output disturbance*

$\mathbf{P_a} = diag\left\{ \begin{array}{cccc} 0.1 & 0.1 & 0.1 & 0.1 \end{array} \right\}$

$\mathbf{Q_a} = diag\left\{ \begin{array}{cccc} 0.002 & 0.002 & 0.002 & 0.002 \end{array} \right\}$

$\mathbf{R_a} = diag\left\{ \begin{array}{cc} 0.001 & 0.001 \end{array} \right\}$

3. *Model augmented for unknown model parameters*

$\mathbf{P_a} = diag\left\{ \begin{array}{cccccccc} 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \end{array} \right\}$

$\mathbf{Q_a} = diag\left\{ \begin{array}{cccccccc} 0.002 & 0.002 & 0.002 & 0.002 & 45 & 0.002 & 0.002 & 0.002 \end{array} \right\}$

$\mathbf{R_a} = diag\left\{ \begin{array}{cc} 0.001 & 0.001 \end{array} \right\}$

4. *Model augmented for unknown model parameters and output disturbance*

$\mathbf{P_a} = diag\left\{ \begin{array}{ccccccccc} 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \end{array} \right\}$

$\mathbf{Q_a} = diag\left\{ \begin{array}{ccccccccc} 0.002 & 0.002 & 0.002 & 0.2 & 0.0002 & 100 & 0.0002 & 0.0002 & 0.0002 \end{array} \right\}$

$\mathbf{R_a} = diag\left\{ \begin{array}{cc} 0.001 & 0.001 \end{array} \right\}$

Addition tuning parameters of ASUKF algorithm

$\alpha = 0.001 \qquad \beta = 200 \qquad \kappa = 0$

Initial state and initial augmented state estimate

$\mathbf{x}\,(0) = \left[\begin{array}{ccc} 0.01 & 0 & 0 \end{array} \right]^T$

1. *Model augmented for output disturbance*

$\hat{\mathbf{x}}_\mathbf{a}\,(0) = \left[\begin{array}{cccc} 0.01 & 0 & 0 & 0 \end{array} \right]^T$

2. *Model augmented for unknown model parameters*

$\hat{\mathbf{x}}_\mathbf{a}\,(0) = \left[\begin{array}{cccccccc} 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right]^T$

3. *Model augmented for unknown model parameters and output disturbance*

$\hat{\mathbf{x}}_\mathbf{a}\,(0) = \left[\begin{array}{ccccccccc} 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right]^T$

5.4.3 Results of ASEKF-NMPC Controller

This subsection is dedicated to presenting the validation results of ASEKF-NMPC controller. The validation results are intended to provide a rough idea of the importance of implementing ASEKF in the formulation of state estimation based NMPC. The validation process is performed in four steps called *validation tests*, each corresponding to a particular set of uncertainties assumed about the system under control.

Validation Test One

A first validation test investigates how well the ball can track a given reference position trajectory under noisy condition. That is, the plant-model mismatch and unknown disturbances are not present. The controller has to estimate the ball's velocity from noisy position and coil current measured outputs to determine an optimal control action.

The closed-loop time responses of magnetic levitation system state variables are shown in Figure 5.16. Initially all state variables are zero except for $x_1(k)$, which has a value of $0.01\,m$. A control voltage increases suddenly to its upper limit, thereby forcefully accelerating the ball towards the reference position $x_1^{sp}(k)$. The time response of coil current is shown in Figure 5.16. The ball's position response in a first subplot shows that ASEKF-NMPC controller is able to drive the ball towards the reference position trajectory and keep it as close as possible. The second subplot shows the ball's velocity estimated by the ASEKF. The last subplot illustrates time response of control voltage.

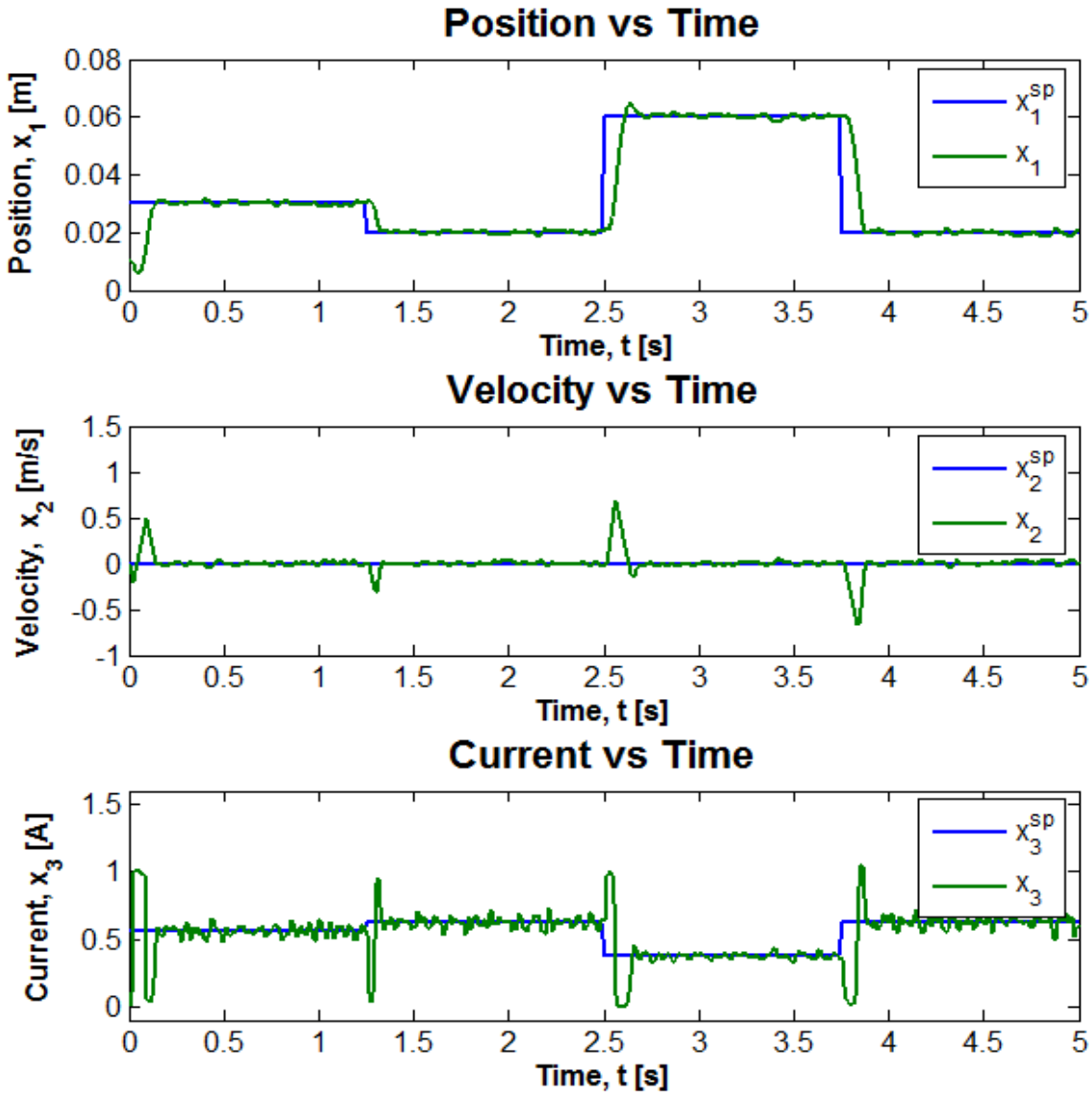


Figure 5.16: ASEKF-NMPC controller - States time responses under noisy conditions.

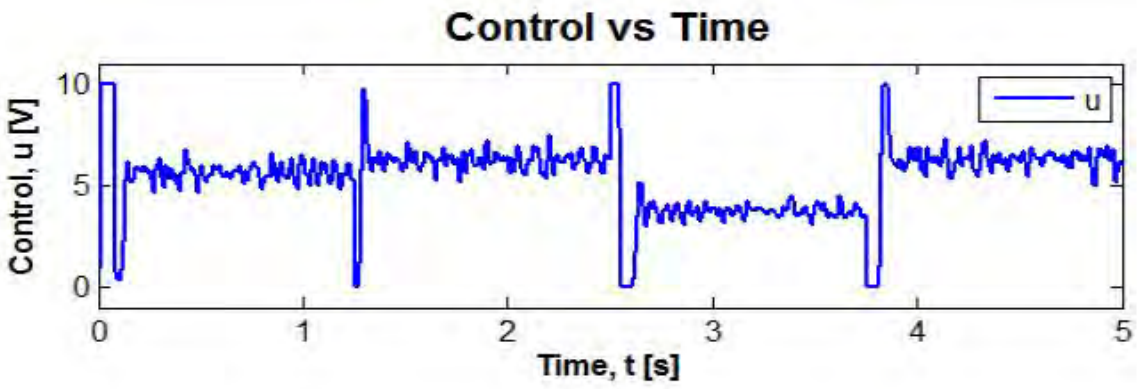


Figure 5.17: ASEKF-NMPC controller - Control input under noisy conditions.

The simulation data in Figure 5.16, in general, confirms that ASEKF produces good estimates for all state variables of magnetic levitation system from the noisy measurements. Hence, the performance obtained with ASEKF-NMPC is robust to noise.

Validation Test Two

The second validation test investigates how well the ball can track a given reference position trajectory when the output step disturbance of an unknown magnitude is imposed on the ball. In this validation test, the ball is subjected to $-0.02\,m$ output step disturbance. Magnetic levitation system model augmented for output disturbances is then implemented for the design of ASEKF estimator in the formulation.

Figure 5.18 displays closed-loop time responses of magnetic levitation system state variables. The first subplot of the figure shows the ball's position within the air-gap. The ball is initially at $0.01\,m$ above the ground position, but it is driven towards the constant reference position trajectory and kept close to it under the influence of control voltage in the face of noise. At the time instant $t = 1.25\,sec$ the output step disturbance is injected to push the ball away from $x_1^{sp}(k)$. The ball's position response shows that ASEKF-NMPC controller is able to satisfactorily reject the disturbance, thereby making the ball to quickly return to $x_1^{sp}(k)$. This indicates that ASEKF is able to estimate the disturbance well and then sent the accurate estimation information to NMPC algorithm to compute appropriate control actions. The last subplot depicts the coil current which is responsible for making the electromagnet to generate enough electromagnetic force to counteract the step disturbance. The second subplot shows the ball's velocity estimated using ASEKF.

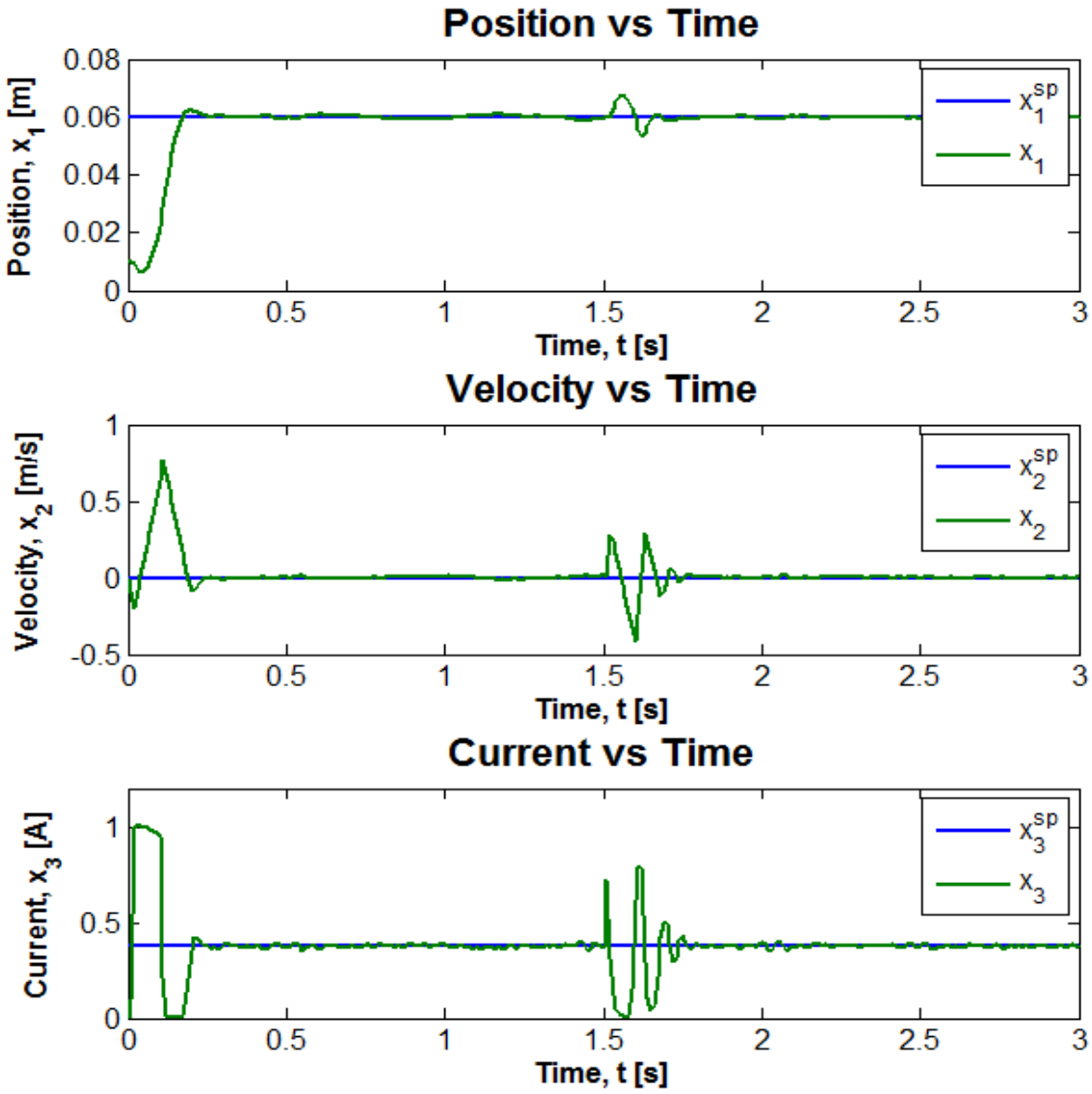


Figure 5.18: ASEKF-NMPC controller - Output step disturbance rejection.

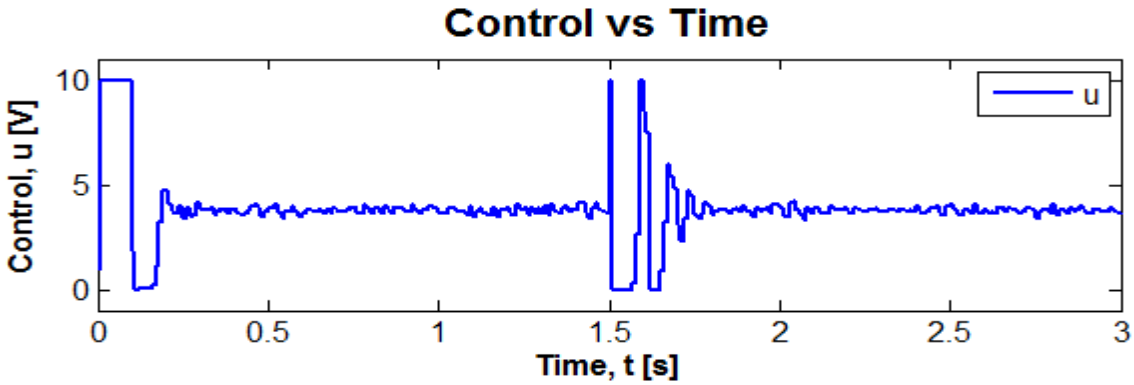


Figure 5.19: ASEKF-NMPC controller - Control input generated to counteract output step disturbance.

Validation Test Three

The first and second validation tests do not take into account the issue of plant-model mismatch which usually occurs in practice. The presence of plant-model mismatch can affect the closed-loop stability and performance of a control system. Therefore, it is important for any practical controller to be relatively robust towards at least moderate levels of plant-model mismatches.

The third validation test is therefore established to investigate how best the ferromagnetic ball can track a given reference in response to 5% parametric model mismatch. Magnetic levitation system model augmented for unknown model parameters is then implemented for the design of ASEKF estimator in the formulation. Notice that along the main diagonal of $\mathbf{Q_a}$, an element corresponding to the coil resistance R is relatively set to a big value. This is done because it was apparent from a series simulation runs that the closed-loop performance of ASEKF-NMPC controller is highly depended on how accurate R is estimate.

Figure 5.20 shows the closed-loop time responses of state variables and control input in response to 5% parametric plant-model mismatch in all parameters of the model. It is clear from the first subplot that the ball achieved good tracking performance in the presence of parametric model mismatch. The control action taken by the ASEKF-NMPC controller is shown in Figure 5.21. The controller seems to be very active and so does the coil current in the third subplot compared to the controller in the first validation test. Possibly this is due to the fact that since the ASEFK in the control formulation is also tasked with estimating modeling parameters, its performance in filtering the measurement noise would deteriorate. Hence why the controller has to work hard in order to keep the ball as close as possible to the reference position.

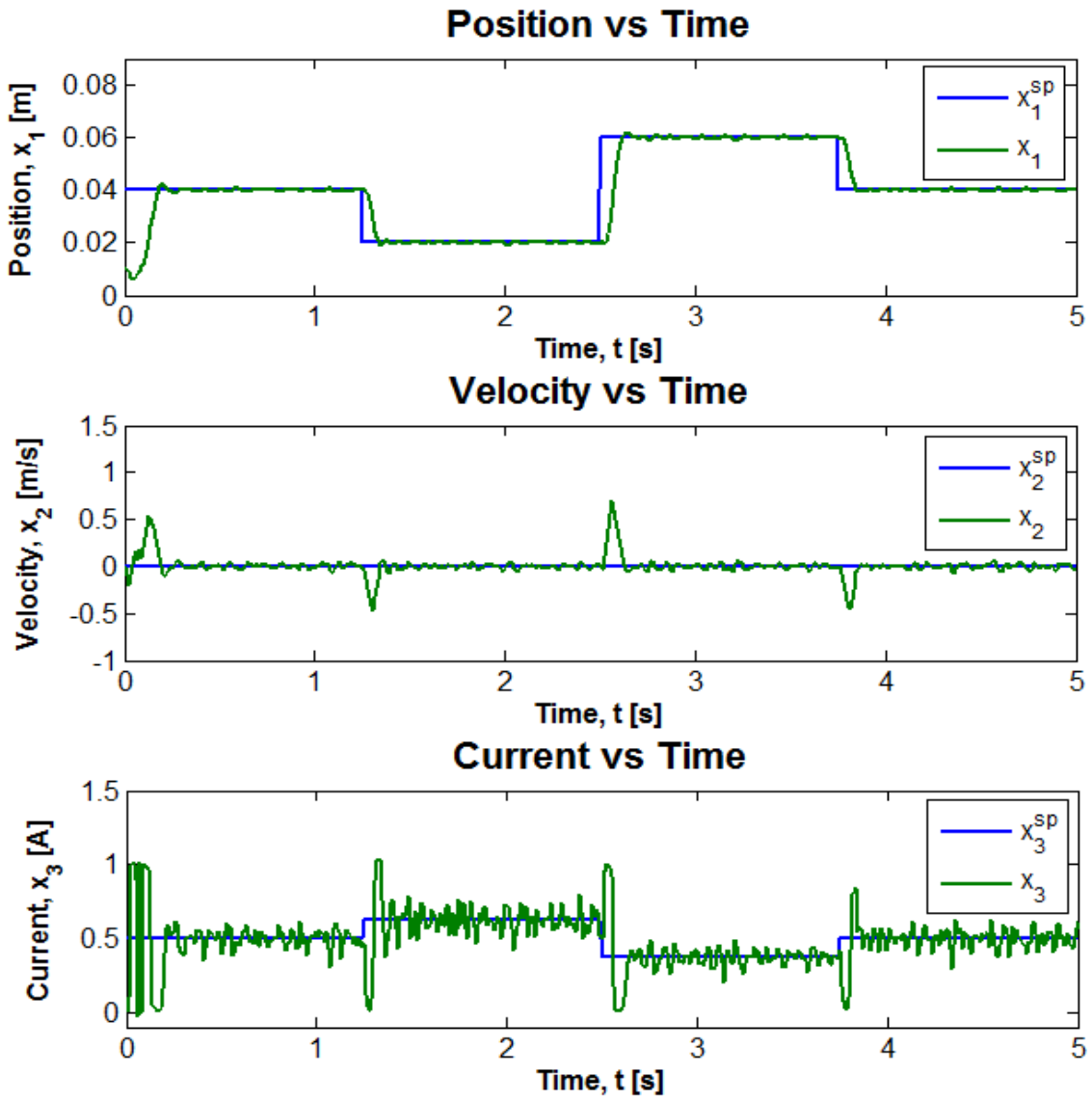


Figure 5.20: ASEKF-NMPC controller - State time responses the conditions of 5% parametric plant-model mismatch and noise.

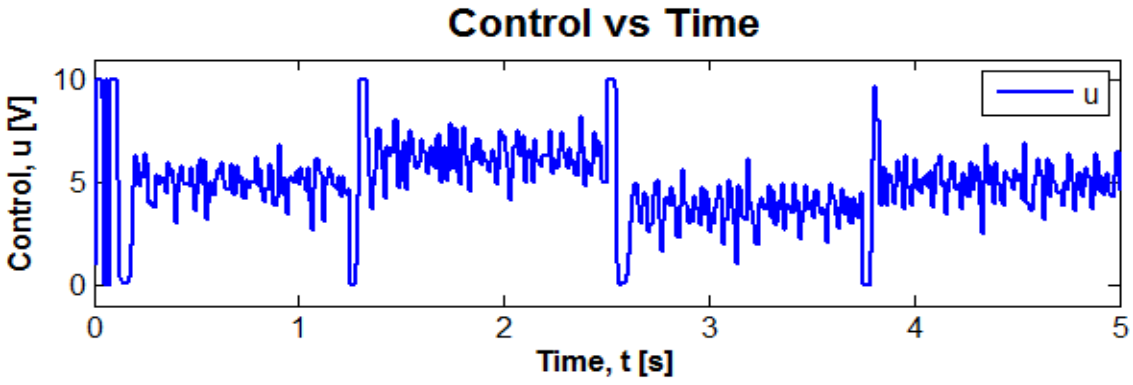


Figure 5.21: ASEKF-NMPC controller - Control input in the presence of 5% parametric plant-model mismatch and noise.

Validation Test Four

The fourth validation test is more challenging and it is the most important. It takes into account the simultaneous existence of both output step disturbance of unknown magnitude and 5% parametric model mismatch in all parameters. It is important in this case to note that in order for ASEKF-NMPC to achieve the best disturbance rejection performance, ASEKF algorithm has to generate accurate estimates for disturbance and model parameters. Therefore, the state space model augmented for output disturbances and model parameters is employed for the design of ASEKF. Noise levels in this case are kept as lower as possible in order to see clearly the effects of model mismatch and disturbance on the closed-loop performance.

The simulation results for the four validation test are provided in Figures 5.22 and 5.31. The closed-loop time responses of three state variables are shown in Figure 5.22. The first subplot shows that the ball is able to track $x_1^{sp}(k)$ though there is a slight mismatch after it has been disturbed. The estimated velocity of the ball as it tracks $x_1^{sp}(k)$ is shown in second subplot and it shows that ASEKF is able to infer unobservable state. The third subplot and Figure 5.31 illustrate the activities of the coil current and control voltage in an effort to keeping the ball close to reference in response to prevailing uncertainties. In general the ASEKF-NMPC controller has achieved good setpoint tracking and disturbance rejection performances.

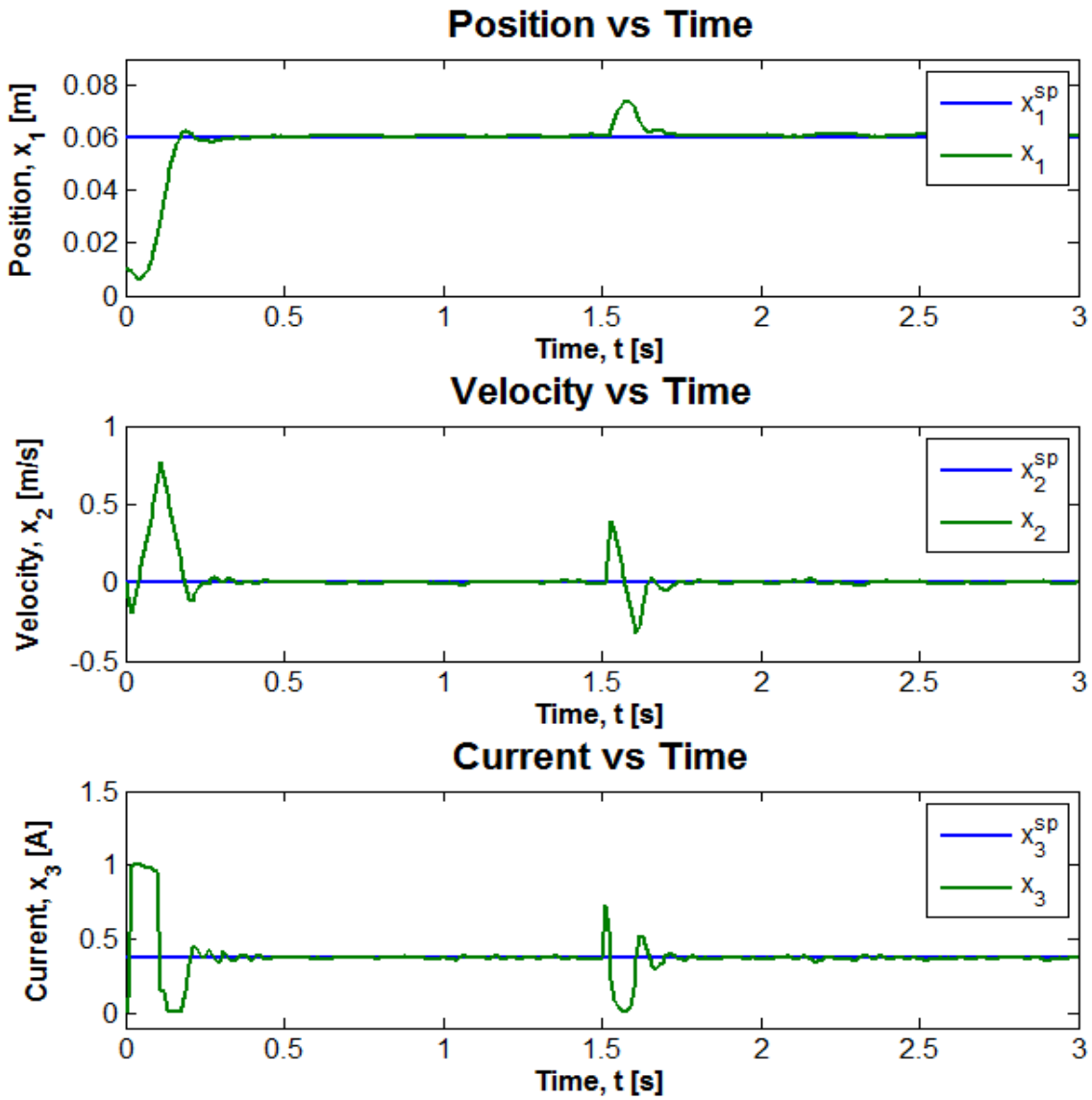
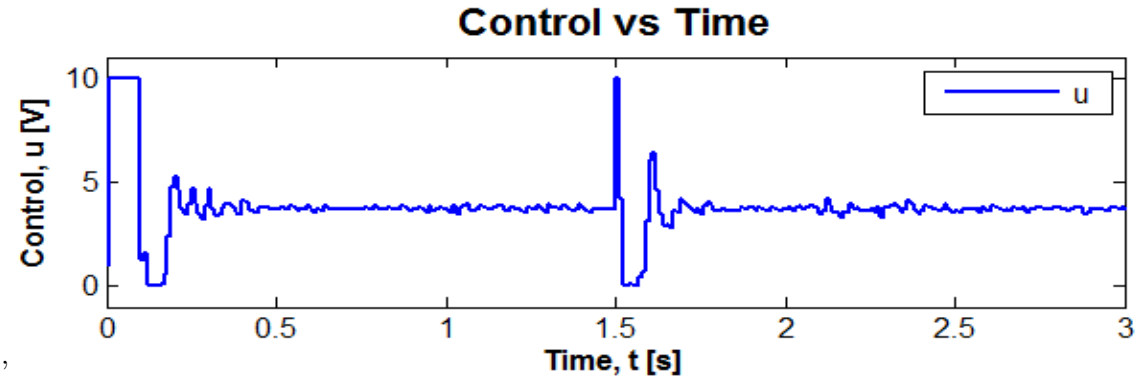


Figure 5.22: ASEKF-NMPC controller - Output disturbance rejection in the presence of 5% parametric plant-model mismatch and noise.



,

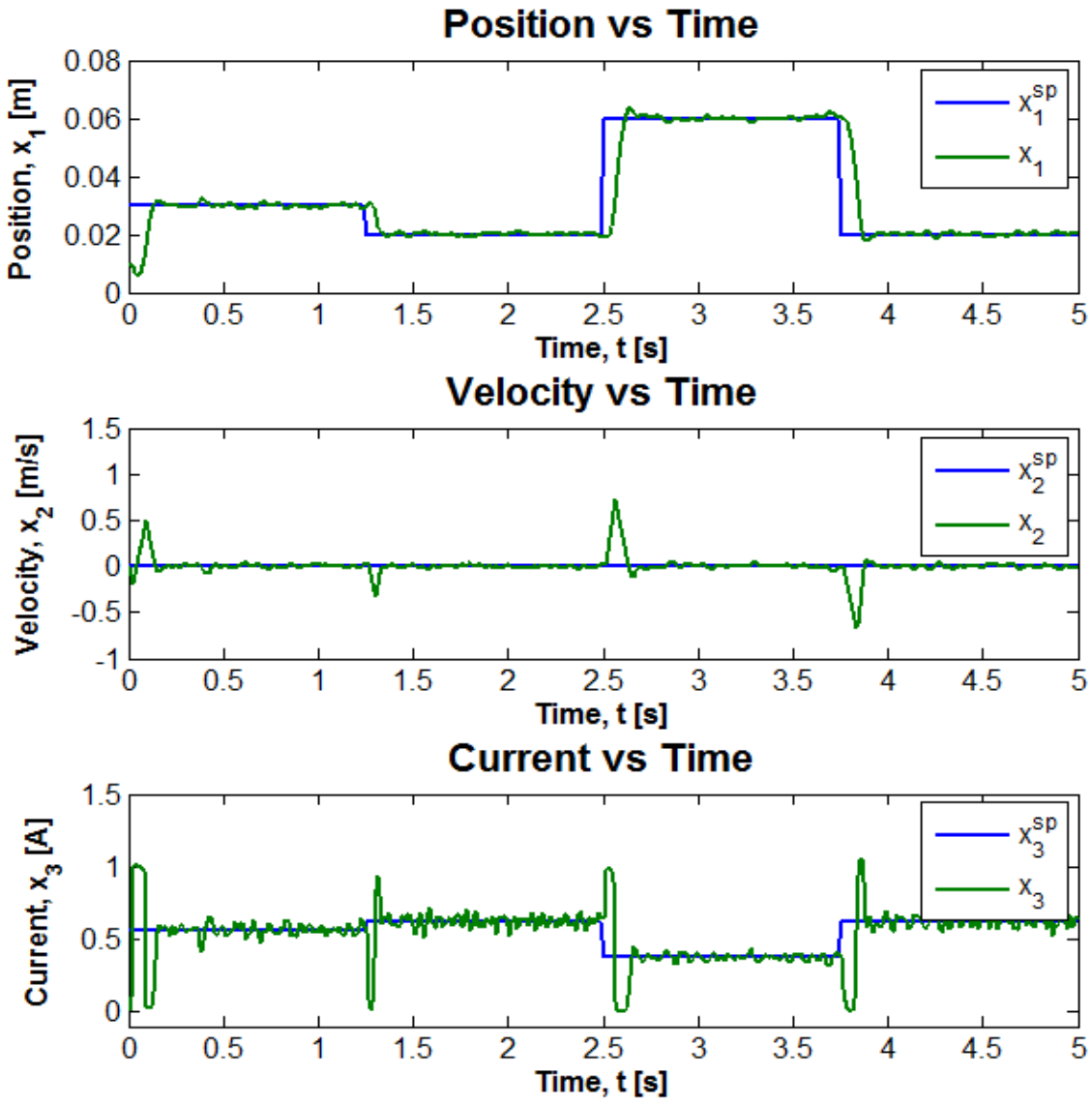
Figure 5.23: ASEKF-NMPC controller - Control input to counteract output disturbance in the presence of parametric plant-model mismatch and noise.

5.4.4 Results with ASUKF-NMPC

The validation results of ASUKF-NMPC controller for a feedback control of magnetic levitation system are presented in this subsection. A procedure for carrying out the validation tests for ASUKF-NMPC is similar to that of ASEKF-NMPC, i.e., the validation process is performed in four similar validation tests.

Validation Test One

The simulation results of the first validation test are depicted in Figures 5.24 and 5.25. The results show that ASUKF-NMPC controller is able to achieve good reference tracking performance in the face of noise. Looking closely at Figures 5.24 and 5.16, it can be realized that in general both ASUKF-NMPC and ASEKF-NMPC controllers have similar performance when implemented for the control of magnetic levitation system under the condition of noise. Hence, the proposed ASUKF-NMPC controller is also robust to measurement noise. The performance of ASUKF-NMPC controller can improve or deteriorate through tuning the addition parameters, α, β and κ.



t

Figure 5.24: ASUKF-NMPC controller - States time responses under noisy conditions.

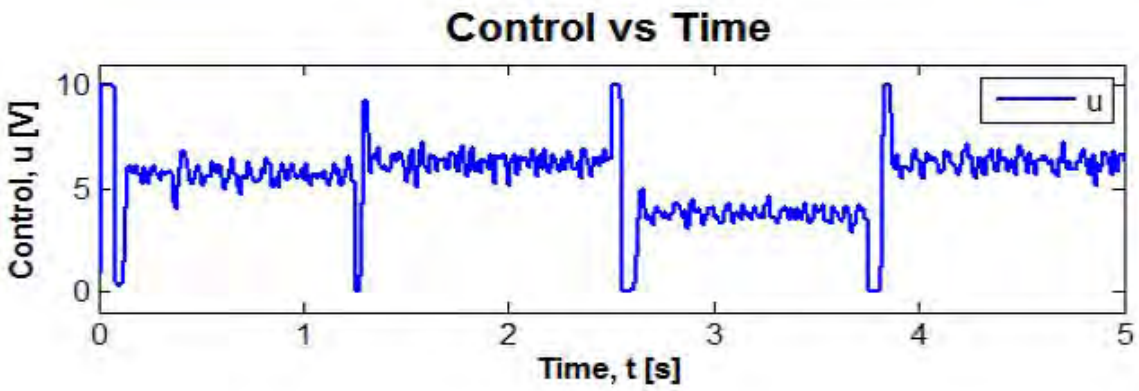


Figure 5.25: ASUKF-NMPC controller - Control input under noisy conditions.

Validation Test Two

In this validation test, the model of the magnetic levitation system is augmented for output disturbances and used for the design of the ASUKF subsystem of the ASUKF-NMPC controller. The controller is then validated for output disturbance performance.

Figures (5.26) and 5.27 provide simulation results of the validation test. Figure (5.26) shows that ASUKF-NMPC controller is able to achieve good setpoint tracking and disturbance rejection performances. This is the indication that ASUKF in the controller is able to provide the NMPC subsystem with accurate state and disturbance estimates to compute appropriate control input to counteract the effects of the disturbance on the closed-loop performance of ASUKF-NMPC. The time response of the control input is shown in Figure 5.27. The output disturbance rejection performance of the ASUKF-NMPC controller is more or less similar to that of ASEKF-NMPC controller.

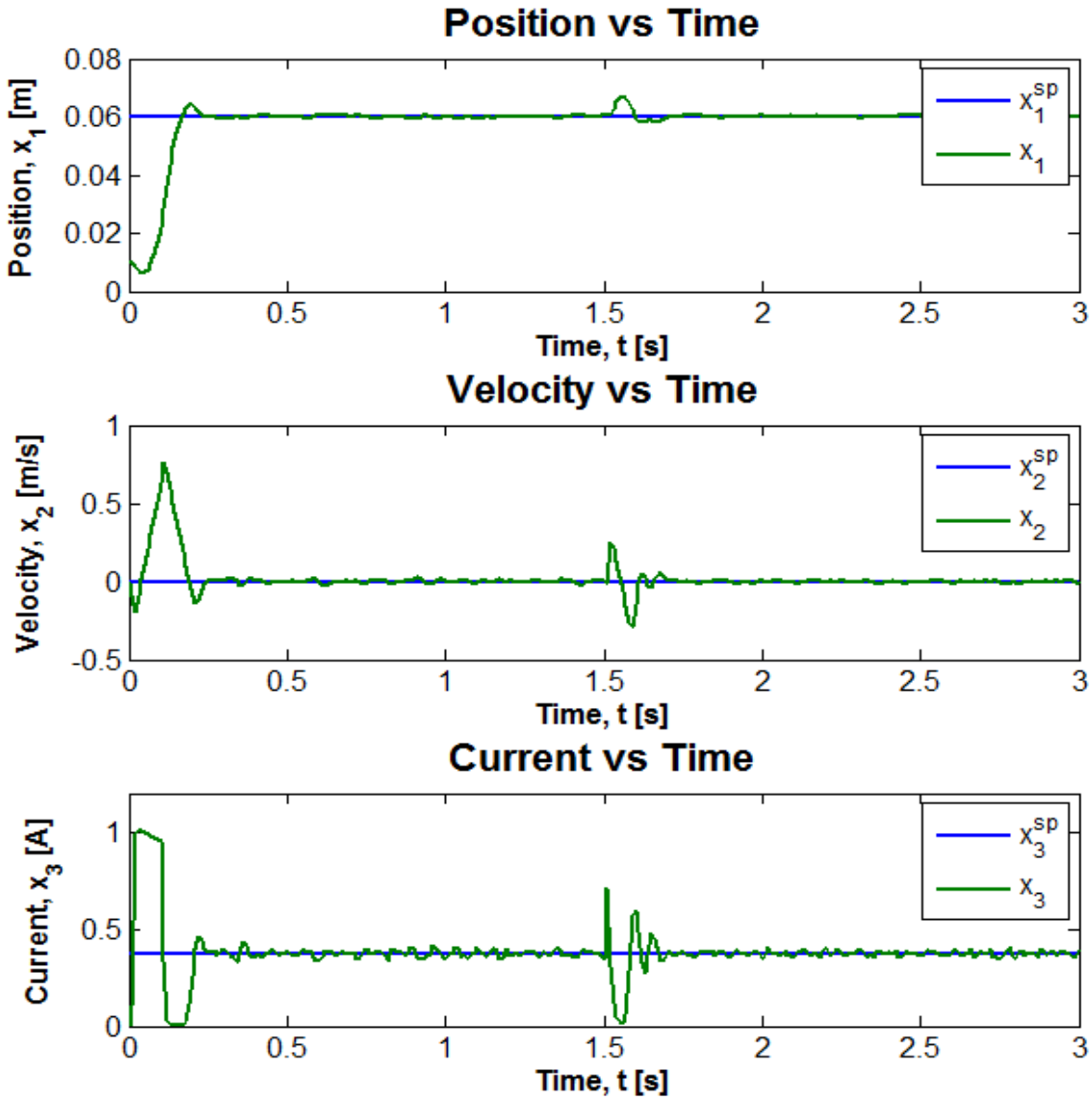


Figure 5.26: ASUKF-NMPC controller - Closed-loop time responses of state variables and control input in the presence of output step disturbance.

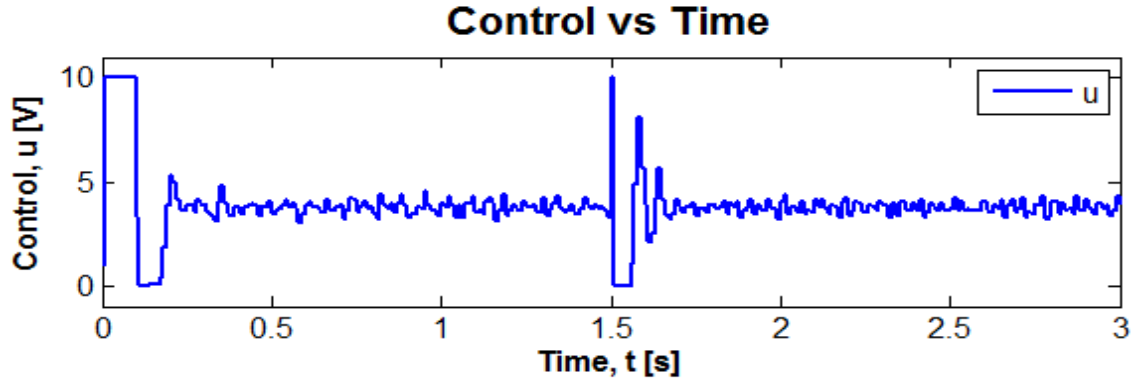


Figure 5.27: ASUKF-NMPC controller - Disturbance estimation using ASUKF for state estimation based NMPC.

Validation Test Three

The performance of ASUKF-NMPC controller in the presence of parametric plant-model mismatch is established in the third validation test. In this validation test, the model of the magnetic levitation system is augmented for unknown model parameters and used for the design of the ASUKF subsystem of the ASUKF-NMPC controller.

The simulation results of the closed-loop performance are depicted in Figures 5.28 and 5.29. It is clear from the first subplot of Figure 5.28 that ASUKF-NMPC controller achieves good reference tracking performance in the face of parametric plant-model mismatch. The third subplot and Figure 5.29 show the coil current and control input voltage, respectively, as they adjusted accordingly to make electromagnet to generate appropriate electromagnetic force to keep the ball close to $x_1^{sp}(k)$. The second subplot of Figure 5.28 shows the ball's velocity estimated using ASUKF estimator.

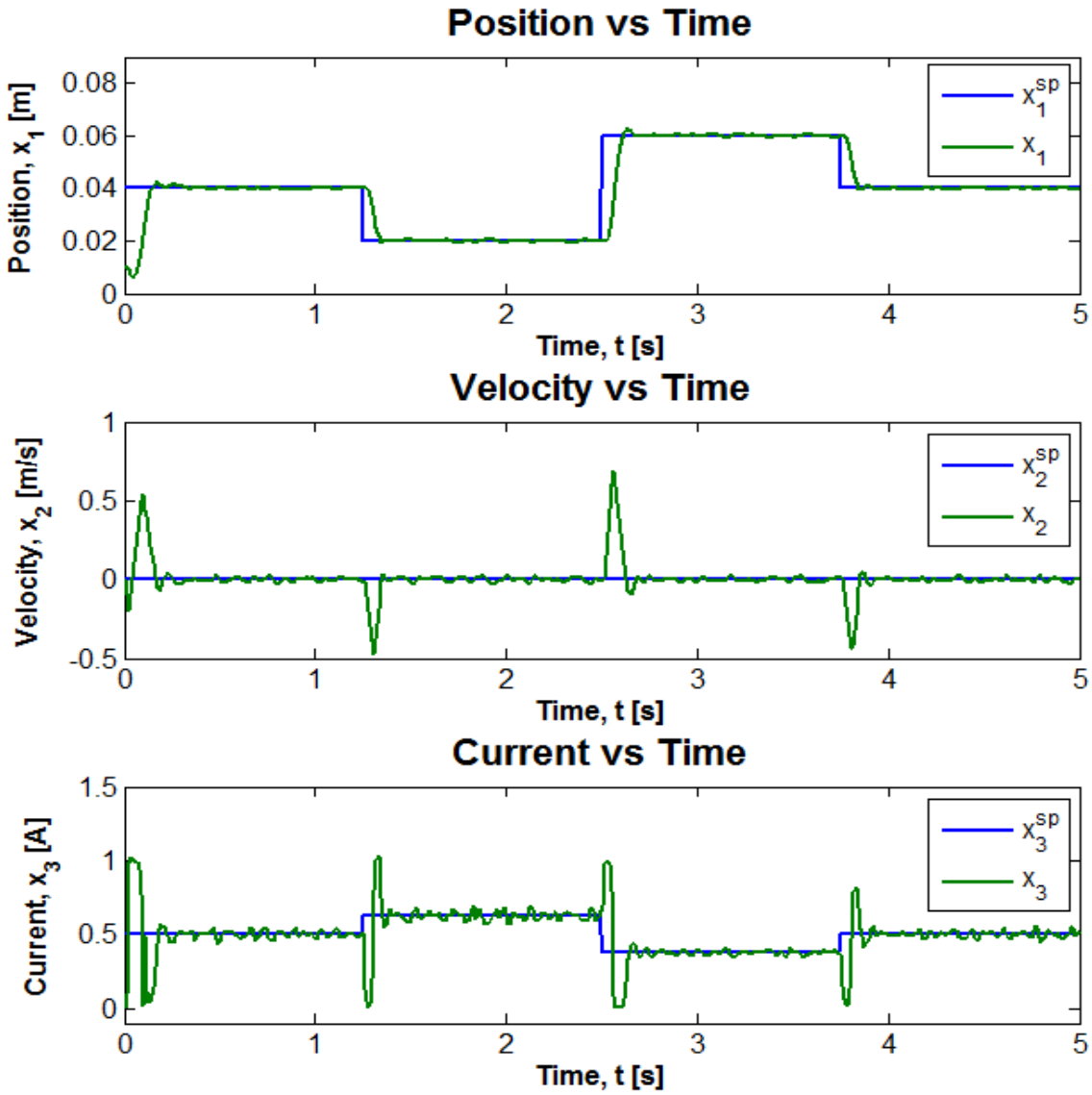


Figure 5.28: ASUKF-NMPC controller - State time responses the conditions of 5% parametric plant-model mismatch and noise.

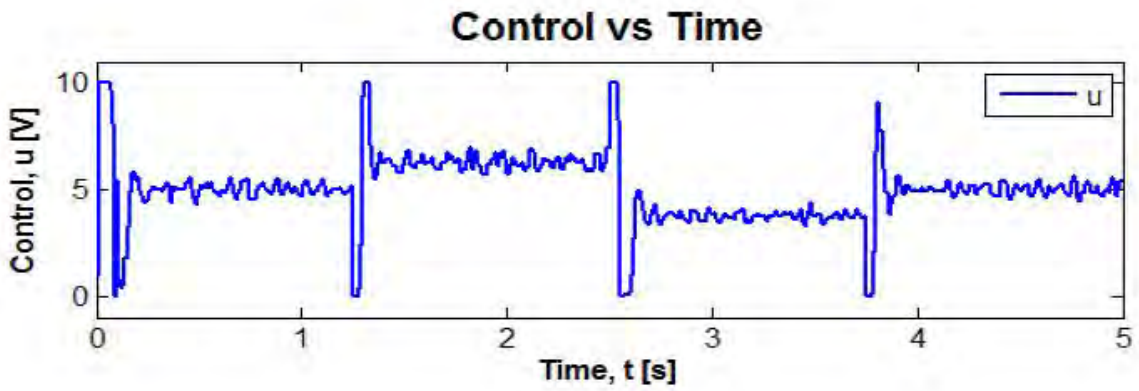


Figure 5.29: ASUKF-NMPC controller - Control input in the presence of 5% parametric plant-model mismatch and noise.

Validation Test Four

The fourth validation test is intended to validate the performance of the proposed ASUKF-NMPC controller for handling the output disturbance and parametric plant-model mismatch simultaneously. Therefore the model of the magnetic levitation system is augmented for both unknown output disturbance and unknown model parameters and then used for the design of ASUKF estimator in ASUKF-NMPC controller.Noise levels in this case are kept as lower as possible in order to see clearly the effects of model mismatch and disturbance on the closed-loop performance. α, β and κ are retuned through trial and error and in this case they are set to the values given as follows,

$$\alpha = 0.0001, \qquad \beta = 2000, \qquad \kappa = 0. \tag{5.91}$$

Figures 5.30 and 5.31 show the simulation results of the control of magnetic levitation system using ASUKF-NMPC controller in response to parametric plant-model mismatch and output step disturbance. The control voltage shown in Figure 5.31 is adjusted accordingly to drive the ferromagnetic ball towards $x_1^{sp}(k)$ and keep it as close as possible. The second subplot of Figure 5.30 provides the estimated ball's velocity as it moved towards the reference. The time response of ball's position in the first subplot make it apparent that ASUKF-NMPC controller is able to keep the ball in the vicinity of $x_1^{sp}(k)$. When the output step disturbance is injected into the position of the ball at $t = 1.25\,sec$, the ball drifted away from the reference for sometime and then return to it quickly.

This indicates that ASUKF-NMPC is able to handle plant-model mismatches and output disturbances simultaneously thereby making it more interesting to be applied for the control of systems in which output disturbances and plant-model mismatches exist simultaneously.

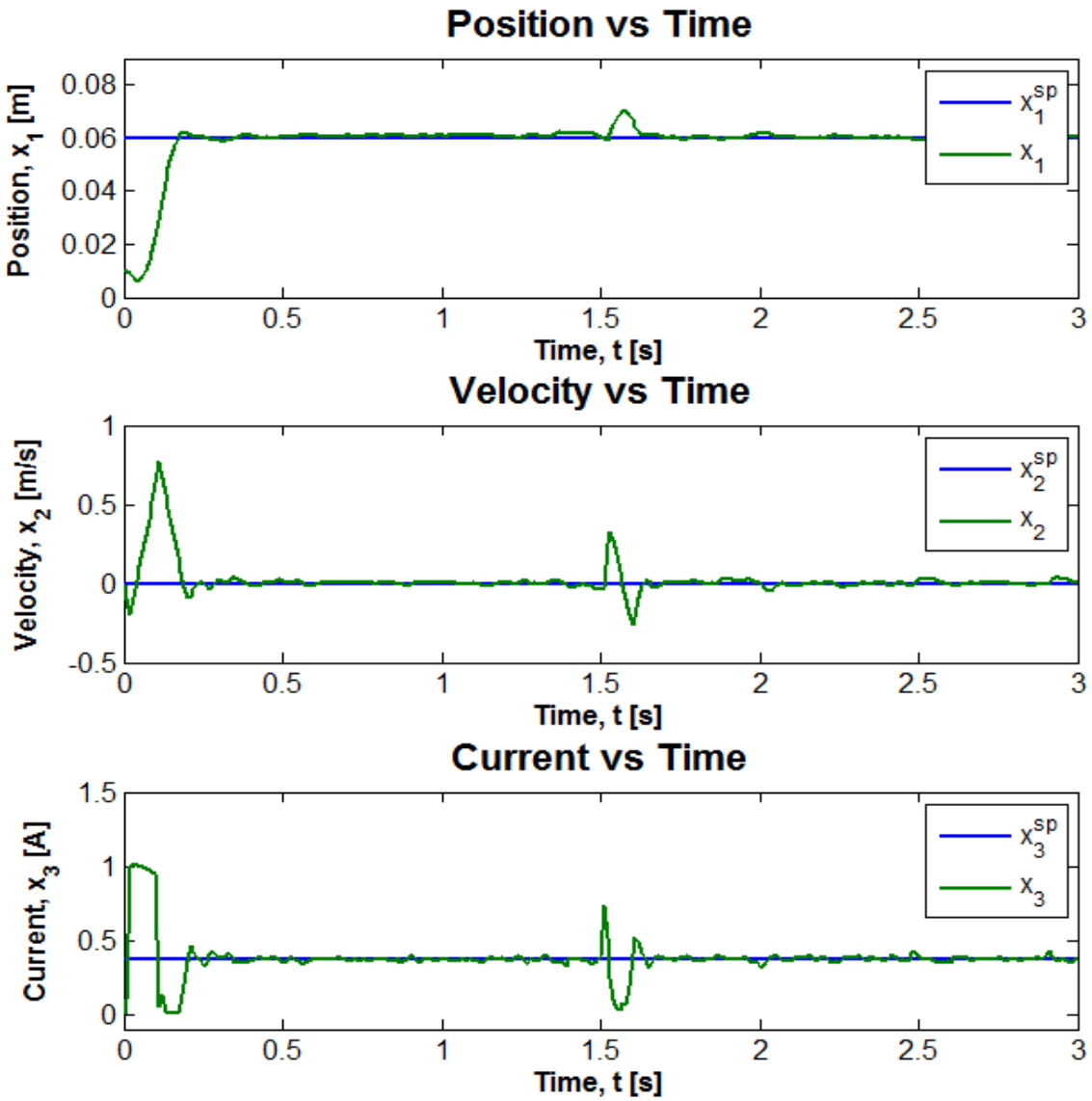


Figure 5.30: ASUKF-NMPC controller - Output disturbance rejection in the presence of 5% parametric plant-model mismatch and noise.

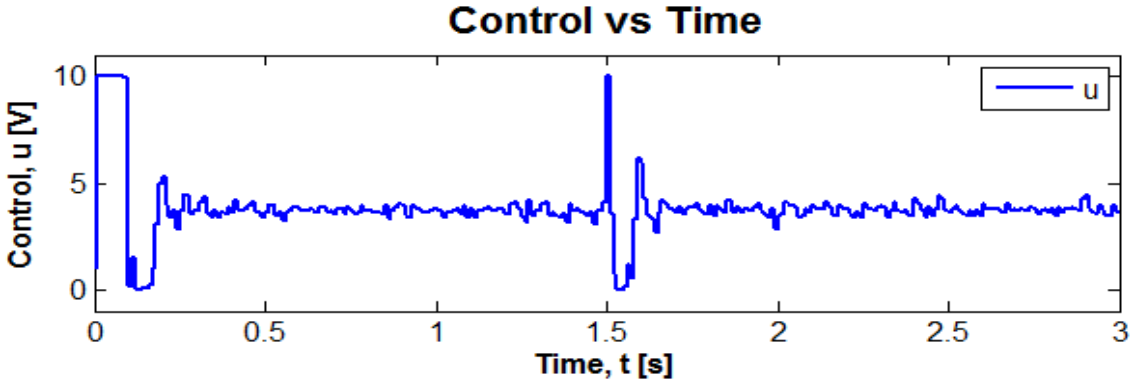


Figure 5.31: ASUKF-NMPC controller - Control input to counteract output disturbance in the presence of parametric plant-model mismatch and noise.

Chapter 6

Multiobjective Optimization

When designing state estimation based control system, it is always important to determine a state estimation method which best addresses the design objectives. However, it may be difficult for the designer to compare state estimation methods solely by their theoretical properties. Usually they have to be implemented on the system and then establish an empirical comparison or evaluation based on the quality of results obtained.

In most cases, the evaluation problem of algorithms consists of two or more (often conflicting) design objectives. To ensure that evaluations are not biased in this situation, the evaluation problem needs to be formulated as a *multiobjective optimization* (MOO) problem, whereby the conflicting design objectives are accurately represented by their respective objective functions. MOO has a good history of success in many fields of science including control engineering, in which trade-off decisions have to be taken. They have been utilized to determine optimal solutions to many problems of many conflicting objective functions [27], where optimality is defined in terms of the trade-offs between the computed objectives and concept of *Pareto Efficiency* [28, 29].

A Pareto Efficient situation is one where an improvement in one or more objective function(s) is impossible without causing a deterioration in at least one or more other objective functions [29]. The objective functions are either minimized or maximized to generate the trade-off solutions referred to as Pareto optimal solutions. A vector of decision variables corresponding to solutions included in the Pareto optimal set is called a *non-dominated* vector and a plot of the objective functions whose non-dominated vectors are in the Pareto optimal set is referred to as a *Pareto front* [28, 29].

This chapter reviews some important topics in multiobjective optimization. Firstly, multiobjective optimization design is reviewed under which the concepts such as the Pareto optimal and approximation of the Pareto front using Non-dominated Sorting Genetic Algorithm (NSGA-II) are discussed. Secondly, an importance of using *Level Diagrams* for visualization and analysis of the Pareto sets is looked into.

6.1 Multiobjective Optimization Design

In this section, the author presents a short review of the multiobjective optimization problem. As the name suggests, the multiobjective optimization problem is any optimization problem which is made up of two or more objectives. To describe it mathematically, consider given a vector of inputs or decision variables

$$[\theta_1, \cdots, \theta_i] \in \Omega \tag{6.1}$$

where $(\theta_1, \cdots, \theta_i)$ is a feasible set of i decision variables and Ω is the decision space. The multiobjective optimization problem can then be written mathematically as

$$\underset{\theta \in \Omega}{min} \quad J(\theta) = [J_1(\theta), \cdots, J_k(\theta)] \tag{6.2}$$

where $J(\theta)$ is a vector of objective functions and $k \geq 2$ is the number of objective functions. Without loss of generality, the multiobjective problem in this case is formulated as a minimization problem. That is, the objective functions are minimized simultaneously and result in a set of mutually optimal solutions:

$$[\theta_1^*, \cdots, \theta_i^*] \in \Omega_p \tag{6.3}$$

in which no solution(s) is dominant over others. In other words, there is no solution which is considered better than other solutions in Ω_p. Therefore, trying to improve one or more objective functions results in at least one or more objective functions worsening [27]. If some objective functions in (6.2) need to be maximized, those objective functions will need to be multiplied by negative and then minimized with the remaining objective functions.

6.1.1 Pareto Optimality

Usually, the multiobjective optimization problem does not have a solution which minimizes all its objective functions simultaneously. Rather, a set of solutions which when improved in one objective function results in at least one of the other objective functions degrading is feasible. Mathematically, the feasible solution $\theta^* \in \Omega_p$, is said to be Pareto optimal (i.e., dominates another solution) if and only if there exists no other feasible solution, $\theta \in \Omega_p$, such that

$$J_n(\theta) \leq J_n(\theta^*)$$

for all $n \in \{1, \cdots, k\}$ and the *Pareto front*

$$J_i(\theta) < J_i(\theta^*)$$

for at least one index $i \in \{1, \cdots, k\}$.

Figure 6.1 illustrates the concept of optimal solution where f_1 and f_2 are the objective functions to be minimized. Looking at the first plot, all solutions lying in the interior of

the region shaded in "gray" are dominated by the solutions lying on the curve forming a lower boundary of the region. The region is called *dominated feasible region* and it is sub-optimal since both f_1 and f_2 can be reduced further. The solutions lying in the interior of the dominated feasible region are referred to as a *dominated solutions* θ and they are denoted by symbol "$\times$" in the second plot. Furthermore, all solutions lying on the boundary of the dominated feasible region are not dominated by any other solutions. The boundary is technically called the *Pareto front*. The solutions lying on the Pareto front are referred to as a *non-dominated solutions* θ_P and they are denoted by "dot" symbol in the second plot. The Pareto optimal set is defined by

$$\theta_P = \{\theta \in \Omega \,|\, \neg\exists \theta^* \in \Omega \,:\, \theta^* < \theta\}\,. \tag{6.4}$$

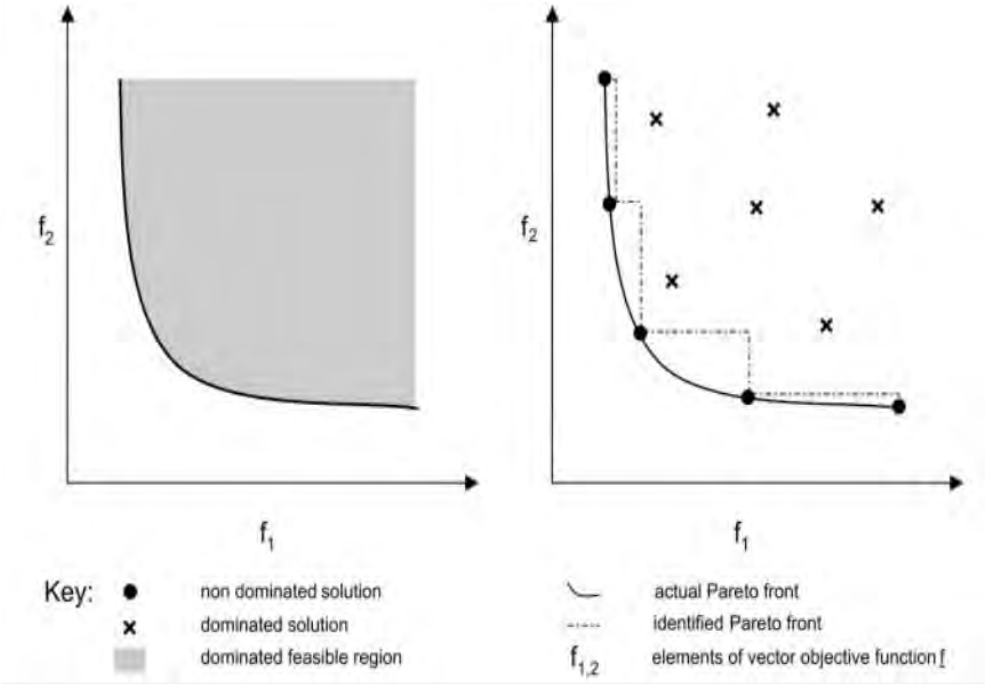


Figure 6.1: The Pareto optimal set [30].

6.1.2 Approximation of Pareto Fronts by NSGA-II

In this thesis, the author chooses to use NSGA-II for evaluations due to its desirable characteristics. The NSGA-II is a fast and elitist multiobjective genetic algorithm as opposed to NSGA algorithm [31]. The comparison of the qualities of solutions in NSGA-II is not only based on their non-domination ranks but also on their crowded distances [32]. Crowding distance provides a measure of the closeness of individual to its neighbors in the population. For instance, if two individuals in the population have similar non-domination rank, then an individual with better crowded distance is selected.

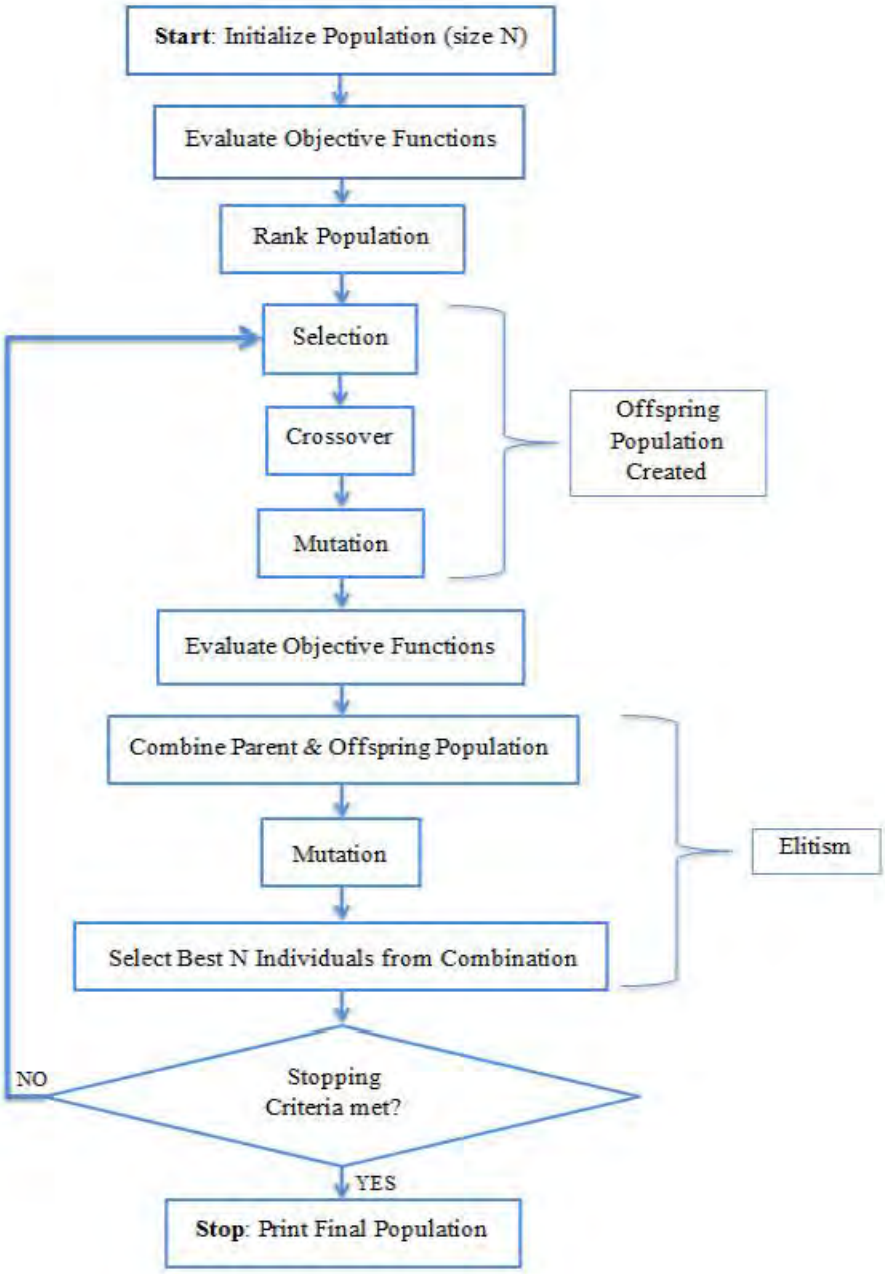


Figure 6.2: Flow diagram illustrating how NSGA-II works

NSGA-II initializes with a population of N individual solutions. Then for each solution $p \in \mathbf{P}$, one has to determine the following:

- The set $\mathbf{S}_p$ which contains all individuals $q \in \mathbf{P}$ which are dominated by individual p. That is, $\mathbf{S}_p = \{q \in \mathbf{P} : p \prec q\}$.
- The integer number n_p which indicates the number of individual solutions which dominate solution p.
- The integer number p_{rank} which indicates a rank of solution p. For instance, $p_{rank} = 1$ when solution p belongs to the first front and $p_{rank} = 2$ when solution p belongs to the second front.
- The *crowding distance* of each solution p which a measure of how close solution p is to

its neighbor solutions. A large average crowding distance results in better diversity in the population $\mathbf{P}$.

After sorting the population, the individual solutions are compared in $\mathbf{P}$ using a *binary tournament selection* with a *crowding-comparison operator* $\prec_n$, based on the following:

- Non-domination *rank*, p_{rank}, of individual solution p. In the tournament, non-dominated solutions with lesser rank are preferred.
- Crowding distance of solution p. If competing solutions in the tournament share the same rank, then the solution with higher crowding distance is preferred.

The following genetic operators are used to produce offspring:

- *Simulated Binary Crossover* (SBX) which simulates the binary crossover observed in nature and it is given by

$$\begin{aligned} c_{1,k} &= \frac{1}{2}\left[\left(1-\beta_k\right)p_{1,k} + \left(1+\beta_k\right)p_{2,k}\right], \\ c_{2,k} &= \frac{1}{2}\left[\left(1+\beta_k\right)p_{1,k} + \left(1-\beta_k\right)p_{2,k}\right], \end{aligned} \tag{6.5}$$

  where $c_{1,k}$ is the i^{th} child with k^{th} component, $p_{i,k}$ is the selected parent and β_k (≥ 0) is a sample from a random number generated having the density given by

$$\begin{aligned} p\left(\beta\right) &= \frac{1}{2}\left(\eta_c+1\right)\beta^{\eta_c},\ 0 \leq \beta \leq 1, \\ p\left(\beta\right) &= \frac{1}{2}\left(\eta_c+1\right)\frac{1}{\beta^{\eta_c+2}},\ \beta > 1, \end{aligned} \tag{6.6}$$

  where η_c is the distribution crossover. This distribution can be obtained from a uniformly sampled random number u between $(0,1)$. That is

$$\begin{aligned} \beta\left(u\right) &= \left(2u\right)^{\frac{1}{\eta+1}}, \\ \beta\left(u\right) &= \frac{1}{\left[2\left(1-u\right)\right]^{\frac{1}{\eta+1}}}. \end{aligned} \tag{6.7}$$

- *Polynomial Mutation* given by

$$c_k = p_k + \left(p_k^u - p_k^l\right)\delta_k, \tag{6.8}$$

  where c_k is the child and p_k is the parent with p_k^u being the upper bound on the parent component, p_k^l is the lower bound and δ_k represents a small variation which is calculated from a polynomial distribution by using

$$\begin{aligned} \delta_k &= \left(2r_k\right)^{\frac{1}{\eta_m+1}} - 1,\ r_k < 0.5, \\ \delta_k &= 1 - \left[2\left(1-r_k\right)\right]^{\frac{1}{\eta_m+1}},\ r_k \geq 0.5, \end{aligned} \tag{6.9}$$

  where r_k is a uniformly sampled random number in the open interval $(0,1)$ and η_m is the mutation index.

The offspring population is combined with the current generation population and selection is performed to set the individuals of the next generation. Since all the previous and current best individuals are added in the population, elitism is ensured [31].

6.2 Level Diagrams Representation of Pareto Fronts

Traditionally, Pareto fronts have been proposed to be represented in 2-D or 3-D plots which are easy to visualize. When Pareto front dimensions increased to a dimension higher than 3-D, it became cumbersome if not impracticable to read useful information from the plots. Owing to that, *Level Diagrams* was proposed as one of the successful tools for visualizing n-dimensional Pareto fronts, preferably for $n > 3$. It provides geometrical visualization of the Pareto front based on a metric distance from an ideal solution which optimizes all objectives simultaneously [33]. Other common n-dimensional visualization tools are *scatter diagrams* and *parallel coordinates,* but they become hard to analyze when dimensions of the data increases, and can require different objective ordering to fully visualize each trade-off [28].

Consider given a vector of decision variables $[\theta_1, \cdots, \theta_i]$ and the vector of objective functions $[J_1(\theta), \cdots, J_k(\theta)]$. The *Level Diagrams* tool is used for the classification of objective functions or decisions variables on the Pareto front based on the proximity to the ideal solution. Each objective function is normalized with respect to its minimum and maximum values on the Pareto front [33, 34]. That is,

$$\bar{J}_i(\theta) = \frac{J_i(\theta) - J_i^m}{J_i^M - J_i^m} \rightarrow 0 \leq \bar{J}_i \leq 1, \tag{6.10}$$

where the subscripts M and m stand for maximum and minimum, respectively. The maximum and minimum values of the objective function are respectively given by

$$J_i^M = \underset{\theta \in \Omega_{P^*}}{min} J_i(\theta), \tag{6.11}$$

and

$$J_i^m = \underset{\theta \in \Omega_{P^*}}{min} J_i(\theta), \quad i = 1, \cdots, s. \tag{6.12}$$

To evaluate a distance to the ideal point, a suitable norm is applied. Different norms can give different views of the characteristics of the Pareto front [33, 34]. Either of the following three norms can be used,

1. 1-norm: $\left\|\bar{J}_i(\theta)\right\|_1 = \sum_{i=1}^{s}\left|\bar{J}_i(\theta)\right|$ with $0 \leq \left\|\bar{J}_i(\theta)\right\|_1 \leq s$,
2. Euclidean norm (2-norm): $\left\|\bar{J}_i(\theta)\right\|_2 = \sqrt{\sum_{i=1}^{s}\bar{J}_i(\theta)^2}$ with $0 \leq \left\|\bar{J}_i(\theta)\right\|_2 \leq \sqrt{s}$,
3. Infinite norm (∞-norm): $\left\|\bar{J}_i(\theta)\right\|_\infty = max\left\{\bar{J}_i(\theta)\right\}$ with $0 \leq \left\|\bar{J}_i(\theta)\right\|_\infty \leq 1$.

The Euclidean norm supplies an accurate evaluation of the geometrical distance to the ideal point, and then offer a better view of the '*real*' shape [33]. On the other hand, infinite norm considers the worst objective (i.e., the farthest from the ideal point) at a specific point [27]. Thus the Infinity norm is commonly used for trade-off analysis between different objectives.

The *Level Diagrams* are plotted as follows: each objective J_i and decision variables θ_i is plotted in a separate 2-D graph with the y-axis corresponding to the value of $\left\|\bar{J}_i\left(\theta\right)\right\|_x$, $x = 1$, 2 *or* ∞ and the x-axis corresponding to the value of the objective or decision variable, in physical units. The variable x represents a norm used. It is important to state that for unbiased interpretation of *Level Diagrams*, each objective or decision variable is represented at the same y position (level) for all graphs, and this means that all information for a single point is drawn at the same position on the y axis for all graphs [34]. This is exemplified by Figures 6.3 and 6.4. See [27] for the details of this example. Figure 6.3 represents a Pareto front in 2-D with two objectives and Figure 6.4 represents the Pareto front in Level Diagrams generated using an Euclidean norm.

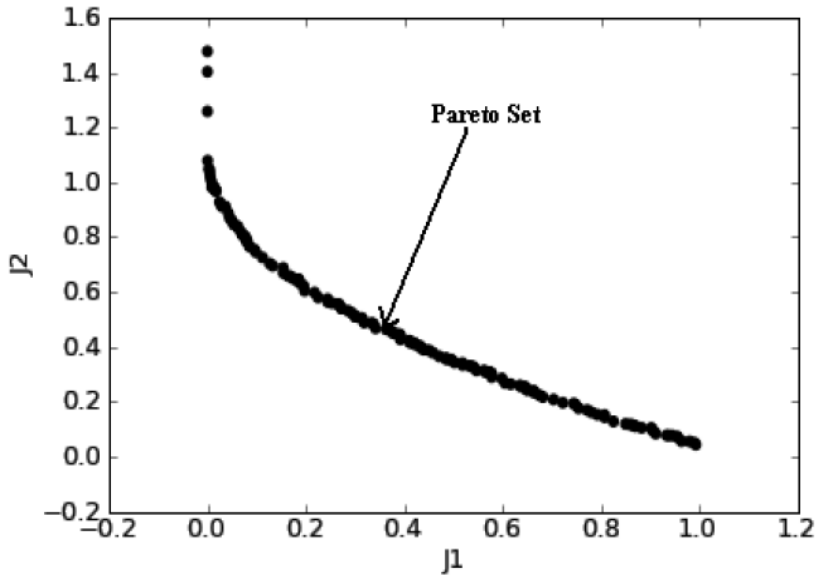


Figure 6.3: Pareto front for two objective functions [27]

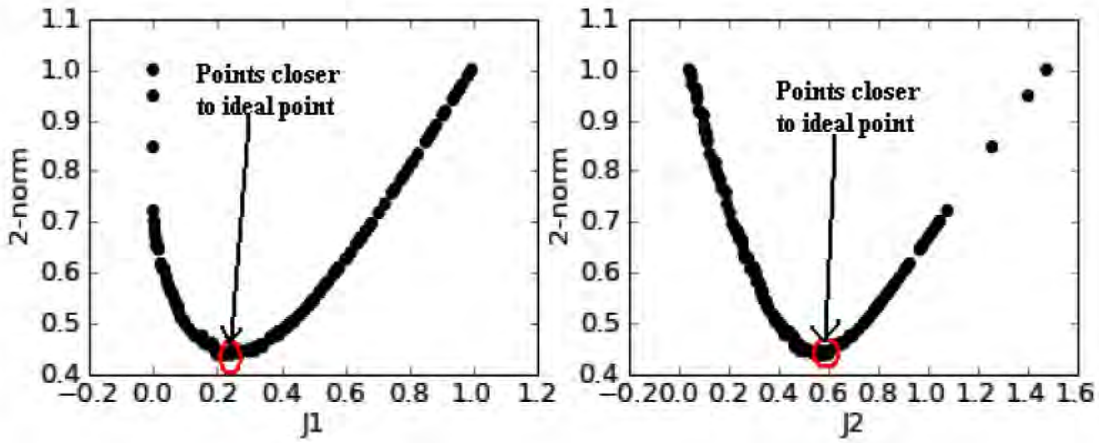


Figure 6.4: 2-Norm *Level Diagrams* [27]

Chapter 7

Case Study: Inverted Pendulum Model

A target system for analyzing the performances of ASEKF and ASUKF methods in the state estimation based NMPC controller is an inverted pendulum on a moving cart. In this chapter, the dynamic equations of the inverted pendulum are derived using Newton's Laws of motion. In order to make the equations implementable on a digital computer, the state space model is derived out of the dynamic equations and then discretized.

7.1 Dynamic Equations

A single inverted pendulum mounted on a moving cart system consists of a cart moving back and forth on a finite track and a pendulum mounted freely on it from a pivot point. The inverted pendulum system is classified as an underactuated system [35]. That is, an external force u is not applied directly to the pendulum, instead, it is applied to the cart and creates a cart displacement from center of a track. Not only does motion of the cart induces motion of the pendulum, but also does a force of gravity acting on the system. A resulting angular displacement of the pendulum is measured from an upright position of the pendulum. The downward position of the inverted pendulum is a stable equilibrium point while the upright position is an unstable equilibrium point. Figure 7.1 illustrates the inverted pendulum mounted on a moving cart in a plane.

Nonlinear dynamics of the inverted pendulum can be derived in different ways, for instance, by applying Lagrangian mechanics or Newton's laws of motion. In this thesis, the nonlinear dynamics is derived by applying Newton's laws of motions. The free body diagrams for the cart and pendulum are shown in Figure 7.2 to help making derivation easier.

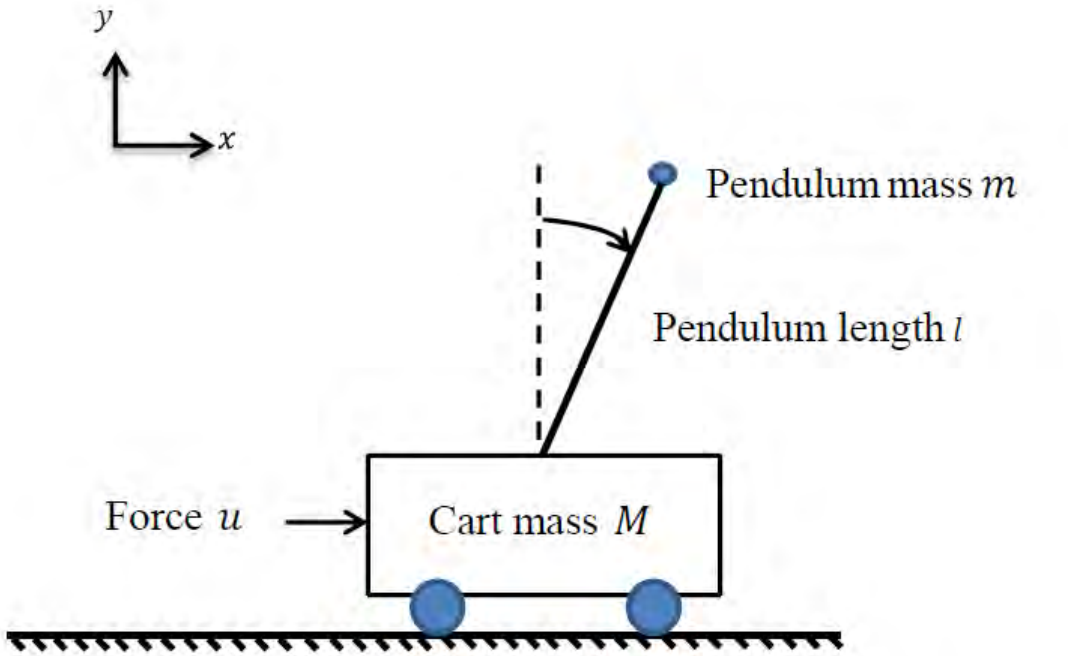


Figure 7.1: Single inverted pendulum mounted on a moving cart.

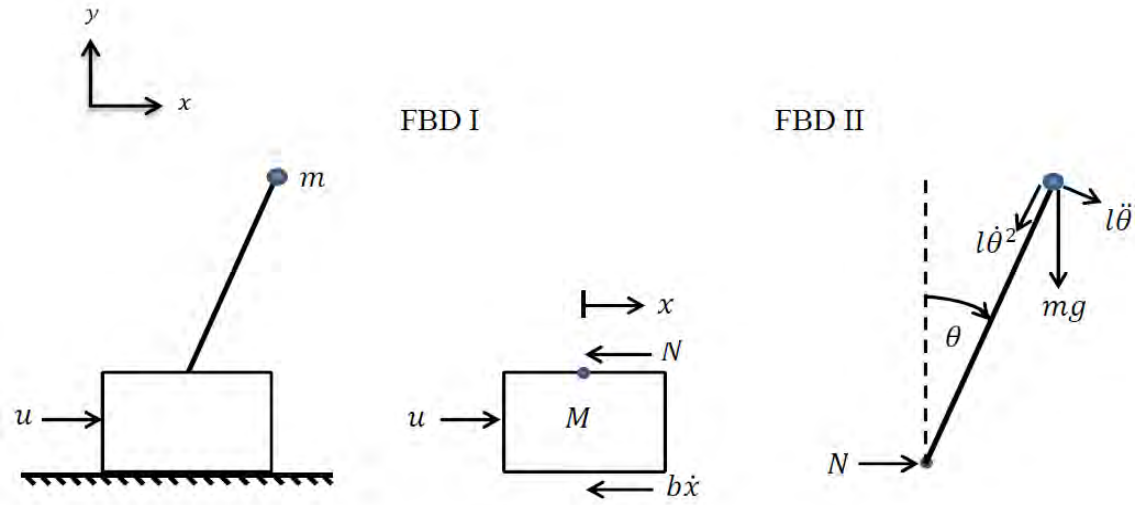


Figure 7.2: The free body diagrams of the single inverted pendulum mounted on the moving cart [3].

The forces acting horizontally (that is, in x-direction) on the cart are summed together. A force balance equation given by Equation (7.1) is a result of the summation of forces. It gives that mass of the cart multiplied by its acceleration must be equal to the external force on the cart minus a friction force, minus the force due to a point mass of the pendulum acting in opposite direction.

$$\sum F_x = u - b\dot{x} - N = M\ddot{x}. \tag{7.1}$$

The forces acting horizontally on the pendulum are summed together in the similar manner. The result of the summation of forces on the pendulum is given by force balance equation given by Equation (7.2). It states that force due to point mass of the pendulum is equal to

point mass multiplied by a total x-directed acceleration of the pendulum. The summation of forces acting perpendicular to the pendulum is given by force balance equation given by Equation (7.3). Equation (7.4) is the torque balance equation.

$$\sum F_x = N = m\left[\ddot{x} + l\ddot{\theta}cos\left(\theta\right) - l\dot{\theta}^2 sin\left(\theta\right)\right], \tag{7.2}$$

$$\sum F_{x^1} = Ncos\left(\theta\right) + mgsin\left(\theta\right) = m\left[l\ddot{\theta} + \ddot{x}cos\left(\theta\right)\right], \tag{7.3}$$

$$\sum \mathcal{T} = -Nlcos\left(\theta\right) = I\ddot{\theta} + h\dot{\theta}. \tag{7.4}$$

Substituting Equation (7.2) into Equation (7.1) results in Equation (7.5) which can be rearranged and simplified into Equation (7.6).

$$M\ddot{x} = u - b\dot{x} - m\left[\ddot{x} + l\ddot{\theta}cos\left(\theta\right) - l\dot{\theta}^2 sin\left(\theta\right)\right], \tag{7.5}$$

$$M\ddot{x} + m\ddot{x} = u - b\dot{x} - ml\ddot{\theta}cos\left(\theta\right) + ml\dot{\theta}^2 sin\left(\theta\right),$$

$$\left[M + m\right]\ddot{x} + b\dot{x} - ml\dot{\theta}^2 sin\left(\theta\right) + ml\ddot{\theta}cos\left(\theta\right) = u. \tag{7.6}$$

In the similarly manner, substituting Equation (7.3) into Equation (7.4) results in Equation (7.7) which can be rearranged and simplified into Equation (7.8).

$$I\ddot{\theta} = -ml\left[l\ddot{\theta} + \ddot{x}cos\left(\theta\right) - gsin\left(\theta\right)\right], \tag{7.7}$$

$$I\ddot{\theta} + ml^2\ddot{\theta} = -ml\ddot{x}cos\left(\theta\right) + mglsin\left(\theta\right) - h\dot{\theta},$$

$$\left[I + ml^2\right]\ddot{\theta} + ml\ddot{x}cos\left(\theta\right) + h\dot{\theta} = mglsin\left(\theta\right). \tag{7.8}$$

The moment of inertia I of a thin rod being rotated about the center of mass is given by Equation (7.9).

$$I = \frac{1}{3}ml^2. \tag{7.9}$$

So substituting (7.9) into Equation (7.8) gives Equation (7.10).

$$\frac{4ml^2}{3}\ddot{\theta} + ml\ddot{x}cos\left(\theta\right) + h\dot{\theta} = mglsin\left(\theta\right). \tag{7.10}$$

Equations (7.6) and (7.10) describe the nonlinear model of the entire single inverted pendulum on a moving cart and they are rewritten in Equations (7.11) and (7.12) to emphasis their importance.

$$\left[M + m\right]\ddot{x} + b\dot{x} - ml\dot{\theta}^2 sin\left(\theta\right) + ml\ddot{\theta}cos\left(\theta\right) = u, \tag{7.11}$$

$$\frac{4ml^2}{3}\ddot{\theta} + ml\ddot{x}cos\left(\theta\right) + h\dot{\theta} = mglsin\left(\theta\right). \tag{7.12}$$

7.2 State Space Model

To numerically simulate the nonlinear dynamic model, Equations (7.11) and (7.12) are put in a state space form given by Equation (2.1). To put Equations (7.11) and (7.12) in state

space form, the equations are first solved for $\ddot{\theta}$ and $\ddot{x}$, respectively. Equations (7.13) and (7.14) are the results of solving for $\ddot{\theta}$ and $\ddot{x}$ in Equations (7.11) and (7.12), respectively.

$$\ddot{\theta} = \frac{mglsin\left(\theta\right) - \frac{ml}{M+m}ucos\left(\theta\right) + \frac{bml}{M+m}\dot{x}cos\left(\theta\right) - \frac{m^2l^2}{M+m}\dot{\theta}^2cos\left(\theta\right)sin\left(\theta\right) - h\dot{\theta}}{\frac{4ml^2}{3} - \frac{m^2l^2}{M+m}cos^2\left(\theta\right)}, \tag{7.13}$$

$$\ddot{x} = \frac{u - b\dot{x} - \frac{3mg}{4}cos\left(\theta\right)sin\left(\theta\right) + \frac{3h}{4l}\dot{\theta}cos\left(\theta\right) + ml\dot{\theta}^2sin\left(\theta\right)}{M + m - \frac{3m}{4}cos^2\left(\theta\right)}. \tag{7.14}$$

To put Equations (7.13) and (7.14) into state space form, the internal variables shown in Equation (7.15) are introduced and substituted in Equations (7.13) and (7.14).

$$x_1 = \theta, \quad x_2 = \dot{\theta}, \quad x_3 = x, \quad x_4 = \dot{x}. \tag{7.15}$$

Equation (7.16) is the result of substituting the internal variables in Equations (7.13) and (7.14) and it is called the state space model of the inverted pendulum on a moving cart.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{mglsin(x_1) - \frac{ml}{M+m}ucos(x_1) + \frac{bml}{M+m}x_4cos(x_1) - \frac{m^2l^2}{M+m}x_2^2cos(x_1)sin(x_1) - hx_2}{\frac{4ml^2}{3} - \frac{m^2l^2}{M+m}cos^2(x_1)} \\ x_4 \\ \frac{u - bx_4 - \frac{3mg}{4}cos(x_1)sin(x_1) + \frac{3h}{4l}x_2cos(x_1) + mlx_2^2sin(x_1)}{M+m-\frac{3m}{4}cos^2(x_1)} \end{bmatrix}, \tag{7.16}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}. \tag{7.17}$$

The Euler method is applied to the model described by Equations (7.16) and (7.17) at the sampling time of $T = 0.01sec$ to find its discrete time counterpart. The resultant discrete time model is used to design the prediction model of the NMPC controller. Furthermore, the discrete time model is augmented for unknown output disturbance and unknown model parameters and then used to design ASEKF and ASUKF subsystems of ASEKF-NMPC and ASUKF-NMPC controllers, respectively.

Table (7.1) provides system parameters of the inverted pendulum on a cart system that is considered here.

Table 7.1: Definitions and typical values for the inverted pendulum.

Parameters	**Symbol**	**Values**	**Unit**
Mass of the Cart	M	3	kg
Mass of the inverted pendulum	m	0.5	kg
Length of the inverted pendulum	l	0.5	m
Sliding friction constant	b	2	Ns/m
Rotational friction constant	h	0.01	Ns/rad
Gravitational constant	g	9.81	m/s^2

Figure 7.3 shows the time responses of the outputs of the model developed above when it is allowed to move freely. The angular position of the pendulum is initially at 15^0 from the vertical position while the cart position is initially in the middle of the rail, i.e., $0\,m$. From the figure it can be seen that the behavior of the model is oscillatory under no influence of external force. As the time progresses, the oscillatory response is damped until there are no more oscillations due to the effects of friction on both the pendulum and the cart.

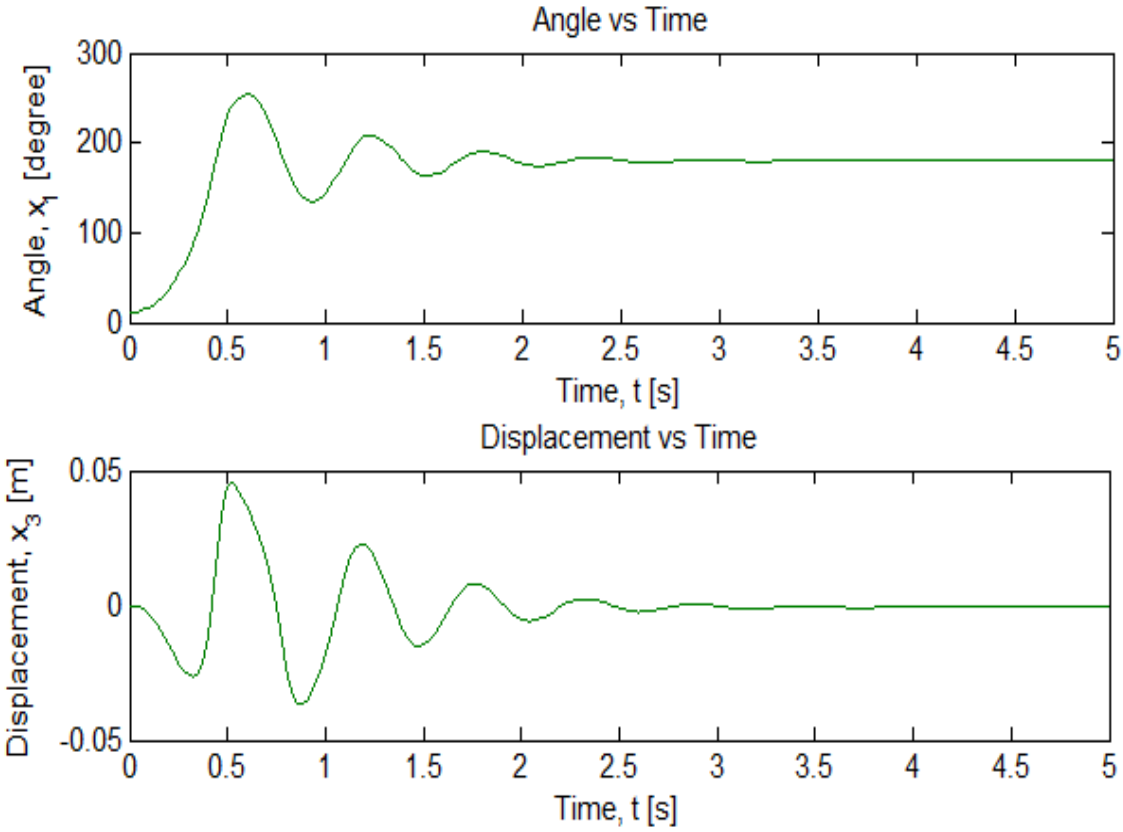


Figure 7.3: Natural behaviour of inverted pendulum on a cart system.

Chapter 8

Evaluation Procedure and Results

This chapter focuses on position control of the inverted pendulum on a moving cart using ASEKF-NMPC and ASUKF-NMPC controllers in order to analyze the closed-loop performances of ASEKF and ASUKF. The analysis is done on the basis of the theory presented in previous chapters. For analysis purposes, parameters of the model which is used to design both estimators and NMPC controller and the model that represents the plant are not exactly the same. That is, there is a parametric plant-model mismatch between the two models. 5% mismatch is used in this thesis. The objective is to investigate whether the estimators in the NMPC formulations do improve the performance of the NMPC so that it swings the pendulum up and stabilizes it in the upright position in the presence of plant-model mismatch. Furthermore, the output step disturbance is introduced to perturb the pendulum from the upright position. The objective is to investigate if the estimators in the controller formulations do improve the capability of the NMPC to reject the disturbance. For the NMPC to achieve these performances, an accuracy of the information received and processed by the controller to compute good control efforts for the pendulum is of utmost importance. Considering how fast the pendulum swings up or recover from disturbances and how much control effort is required to do that are also important to investigate.

The author generates Pareto fronts for the ASEKF-NMPC and ASUKF-NMPC controllers from which the closed-loop performances of ASEKF and ASUKF estimators are analyzed. Furthermore, the author demonstrates how the data from the Pareto fronts can be used to tune the ASEKF-NMPC and ASUKF-NMPC controllers. The time responses of the controllers are presented and compared through simulation runs. The *Level Diagrams* are used to aid visualization of Pareto fronts to facilitate easier analysis.

NSGA-II algorithm is used in this thesis to facilitate the comparative analysis of ASEKF and ASUKF estimators. It employes various objective functions yet to be defined in order to generate approximate Pareto fronts. The author first describes the procedure which was followed to carry out the analysis and later in this chapter, the results of the evaluations are presented and discussed.

8.1 Evaluation Procedure

Here the objective functions needed are specified according to the controller designer's preferences in the first subsection. In the second subsection, decision variables are presented as well as their ranges. The last subsection states how the NSGA-II optimizer was configured for the evaluations.

8.1.1 Objective Functions (Costs)

The cost functions to be used need to be chosen carefully since they serve as performance indices. The performance indices indicate performance of the algorithm(s) being evaluated. The resulting Pareto fronts depend entirely on the cost functions used [27, 28]. In this thesis, ten objective functions are chosen and they are broadly categorized into *tracking and rejection costs, system's response time costs* and *controller input costs*. The objective functions are further categorized into *transient* and *steady-state costs* depending on the phase of response the system is in.

Tracking and Rejection Costs

Two cost functions, V_1 and V_2, are chosen such that they take into account an error between the pendulum angle output $\mathbf{y}_1(t)$ and the upright angle position $\mathbf{r}_1(t)$. V_1 is used for evaluating how much error there is between $\mathbf{y}_1(t)$ and $\mathbf{r}_1(t)$ in the transient response behavior. This helps to see if the ASEKF-NMPC and ASUKF-NMPC controllers are capable of swinging the pendulum towards $\mathbf{r}_1(t)$ in the presence of plant-model mismatch. V_2 evaluates how much error there is between $\mathbf{y}_1(t)$ and $\mathbf{r}_1(t)$, when the output step disturbance is introduced in the steady-state. It is helpful in observing the capabilities of the controllers to reject or suppress the effect of the disturbance as well as the mismatch. The two cost functions are written as follows.

$$V_1 = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \left[\mathbf{r}_1(t) - \mathbf{y}_1(t)\right]^2 dt, \tag{8.1}$$

$$V_2 = \frac{1}{t_3 - t_2} \int_{t_2}^{t_3} \left[\mathbf{r}_1(t) - \mathbf{y}_1(t)\right]^2 dt, \tag{8.2}$$

where $\mathbf{r}_1(t) = 0$, and t_1 and t_2 in Equation (8.1) mark the beginning and the end of the transient or swing up response of the pendulum and t_2 and t_3 in Equation (8.2) mark the beginning and the end of the transient response due to the output step disturbance in the steady-state.

At the downward angular position, the pendulum is considered to have the worst Integral Square Error (ISE) since the downward position is farthest from the upright angular position. Since the pendulum is a periodic system and it is symmetric about the vertical line, the author decided to modify Equations (8.1) and (8.2) to account for periodicity and symmetry. The author proposed to use the expression $1 - \left|1 - \frac{\mathbf{y}_1(t)}{\pi}\right|$ in place of $\mathbf{r}_1(t) - \mathbf{y}_1(t)$. This expression has an advantage that it preserves the linearity with respect

to the angle of the pendulum. The modified version of Equations (8.1) and (8.2) is given by Equations (8.3) and (8.4) as follows.

$$V_1 = \frac{1}{t_1 - t_2} \int_{t_1}^{t_2} \left[1 - \left|1 - \frac{\mathbf{y}_1(t)}{\pi}\right|\right]^2 dt, \tag{8.3}$$

$$V_2 = \frac{1}{t_2 - t_3} \int_{t_2}^{t_3} \left[1 - \left|1 - \frac{\mathbf{y}_1(t)}{\pi}\right|\right]^2 dt. \tag{8.4}$$

System's Response Time Costs

The third and the fourth cost functions, V_3 and V_4, are chosen such that they take into account the approximation of settling time of the system response after the perturbations. The settling time is the time it takes the system transient to enter and remain within a 2% band of the final value after the perturbation. It does not take care of the oscillations and the finite offset in the setpoint tracking as long as the offset settles to within 2% of the final value [28]. V_3 is used to approximate the time it takes the pendulum to swing up from the downward position until it enters and remain within the 2% band of the upright position. On the other hand, V_4 is used to approximate the time it takes the pendulum to enter and remain within the 2% band of the upright position after the occurrence of output step disturbance in the steady-state. The cost functions are proposed and written as follows,

$$V_3 = T_{t,2\%}, \tag{8.5}$$

$$V_4 = T_{d,2\%}. \tag{8.6}$$

The method employed in this thesis finds the approximation of the settling time. There are other methods such as using exponential decay curves to the envelope of the simulation data which find a more precise settling time [28].

Peak Overshoot Costs

The fifth and the sixth cost functions, V_5 and V_6, are chosen such that they take into account the approximation of maximum peak overshoot of the system response after the perturbations. V_5 is used to approximate the maximum overshoots of the angle of the pendulum at the swing up phase. On the other hand, V_6 is used to approximate the maximum overshoot of the pendulum after the occurrence of output step disturbance in the steady-state. V_5 and V_6 are written as follows,

$$V_5 = 1 - \left|1 - \frac{\mathbf{y}_1(t)}{\pi}\right|, \tag{8.7}$$

$$V_6 = 1 - \left|1 - \frac{\mathbf{y}_1(t)}{\pi}\right|. \tag{8.8}$$

Estimation Accuracy Costs

The seventh and eighth cost functions, V_7 and V_8, are chosen such that they take into account a deviation of the state estimate from the actual state. The cost function is used

for evaluating estimation capabilities of the ASEKF and ASUKF estimators. The cost functions are quantified by the ISE and they are given by the following equations,

$$V_7 = \frac{1}{t_1 - t_2} \int_{t_1}^{t_2} [\mathbf{x}(t) - \hat{\mathbf{x}}(t)]^2 \, dt, \tag{8.9}$$

$$V_8 = \frac{1}{t_2 - t_3} \int_{t_2}^{t_3} [\mathbf{x}(t) - \hat{\mathbf{x}}(t)]^2 \, dt, \tag{8.10}$$

where $\hat{\mathbf{x}}(t)$ is the state estimate.

Controller Input Costs

The last two proposed cost functions: V_9 and V_{10}, are quantified by *Integral Squared Difference* (ISD) for the system control $\mathbf{u}(t)$. V_9 is chosen such that it provides a measure of the total control effort required by the controller to drive the system's transient until it enters and remains within 2% band of the final value in the presence of parametric plant-model mismatch. In the case of the pendulum to be specific, it provides a measure of the control effort required for the swing up from the downward position to the 2% band of the upright position. V_{10} is used to provide a measure of the control effort required by the controller to reject the output step disturbance $\mathbf{d}(t)$. The two cost functions are given by

$$V_9 = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} [\mathbf{u}(t) - \mathbf{u}_\infty]^2 \, dt, \tag{8.11}$$

$$V_{10} = \frac{1}{t_3 - t_2} \int_{t_2}^{t_3} [\mathbf{u}(t) - \mathbf{u}_\infty]^2 \, dt, \tag{8.12}$$

where $\mathbf{u}_\infty$ is the final value of the control effort.

8.1.2 Tuning Parameters

To achieve the desired performance, parameters of the proposed NMPC controllers have to be tuned properly. Unlike tuning traditional PI controllers, tuning MPC schemes can be difficult [27]. When MPC scheme is coupled with a state estimator, the number of tuning parameters in the control formulation increases and this may increase the difficulty of tuning further. In [27], MOO tuning method for MPC using PDE optimizer is presented. In this thesis, we propose MOO tuning method for nonlinear MPC using NSGA-II optimizer.

The tuning parameters of the controllers are set as the decision variables of NSGA-II. N_c was set constant, i.e., $N_c = 15$. This is because it can easily be selected offline as long as N_p is known. N_p was bounded on the range [15, 30] and it is chosen such that it is greater or equal to N_c. The upper bound was selected based on the fact that tuning N_p beyond 30 time instances could not improve the responses of the controllers any further. The first and third elements of $\mathbf{Q}$ of the OCP were bounded on the range [5, 25] while the third and fourth on [1, 5]. Since $\mathbf{Q}$ is only used to penalize the error, any range can be used depending on the designer's preferences. $\mathbf{R}$ of the OCP was bounded on the range [0.0001, 0.1], but it can also be bounded to within any range, as long as its upper bound is always less than any element of $\mathbf{Q}$. Otherwise the controller would put more emphasis on minimizing the magnitudes of control input than the error which is often not preferred.

Since it was realized that tuning $\mathbf{P_a}$ is relatively not difficult, only $\mathbf{Q_a}$ and $\mathbf{R_a}$ of the state estimators were set as decision variables. Therefore, $\mathbf{P_a}$ was fixed to identity matrix of appropriate dimensions, which is common when initializing any state error covariance. The elements of $\mathbf{Q_a}$ were tuned through 3 scalars, p_6, p_7 and p_8. All scalars were bounded on the range [0.0001, 0.1]. The range was selected under the assumption that the process noise and random variances in the disturbance and parameters do not have much effect on the closed-loop performance. Any range can be chosen depending on designer's preferences. $\mathbf{R_a}$ is usually chosen to be much smaller than $\mathbf{Q_a}$. In this case, the elements of $\mathbf{R_a}$ were tuned through a scalar p_9 which was bounded on the range [0.0001, 0.01].

The additional parameters of ASUKF: α , β and κ, were also tuned. The significance of κ in the ASUKF estimator is to ensure the (semi)positive definiteness of $\mathbf{P_a}$. Therefore it can be set to any non-negative real number, i.e., $\kappa = 0$, which is a default value. α was bounded on the range [0.001, 0.01]. These bounds were determined through simulation runs. It was observed that any values outside this range lead to fast divergence and sometimes a break of the ASUKF. β was bounded on [100, 500]. Through simulations, the values below $\beta = 20$ lead to quick divergence and break of the ASUKF. So the author decided to set the lower bound to a bigger value in order to eliminate any unforeseeable chances of divergence. The values larger than 500 could be chosen but they do not improve the performance of the ASUKF any further.

8.1.3 NSGA-II Optimizer

The NSGA-II optimizer used in this thesis was developed using Matlab by Seshadri [31]. For the purpose of the evaluations carried out here, the optimizer was run for 50 generations with both crossover rate and mutation rate set to 0.9, and the population size of 80.

8.2 Results

This section presents the *Level Diagrams* results of the ASEKF-NMPC and ASUKF-NMPC controllers for position control of the inverted pendulum on a moving cart. Using these results, the closed-loop performances of ASEKF and ASUKF estimators can be analyzed. Secondly, the section provides the simulation time responses of the controllers by picking the minimum and maximum points on the *Level Diagrams*.

8.2.1 Level Diagram Results

The *Level Diagrams* of the objective functions in the transient and steady-state are shown in Figures 8.1 and 8.2, respectively. Figures 8.3 and 8.4 show the *Level Diagrams* of the decision variables for the Pareto fronts. Note that all the *Level Diagrams* are synchronized using the 2-norm. This is because the 2-norm gives accurate evaluation of conventional geometrical distance to the ideal point and offers a better view of the real shape of the Pareto front in n-D [27, 34].

From the plots a number of observations can be drawn. In Figure 8.1, looking at V_1 and V_3, ASUKF-NMPC is able to achieve good tracking performance at the minimum

control efforts compared to ASEKF-NMPC. Possibly, that is because integrating ASUKF estimator into ASUKF-NMPC formulation provides relatively good state estimation performance compared to ASEKF in ASEKF-NMPC formulation. See V_9. If minimizing the tracking and magnitudes of control inputs simultaneously is desired by the designer, then ASUKF-NMPC provides more chance of obtaining minimum values for both objectives. However, looking at V_5 and V_7, ASUKF-NMPC has more points towards the upper bounds. That means that ASEKF-NMPC is able to achieve relatively faster convergence to the upright reference position with less peak overshoots than ASUKF-NMPC. Possibly, that is the consequence of the fact that ASUKF-NMPC applies weak control efforts to drive the pendulum towards the upright position and try to catch it to withing 2% band of the final value.

In general, ASUKF estimator is able to generate relatively good state estimates for the controller. However, ASEKF-NMPC seems to have a point with the 2-norm lower than any other point of ASUKF-NMPC, even though, the difference is small. Therefore, it is still sensible that the designer may choose to implement ASEKF estimator in state estimation-based NMPC to achieve good tracking performance.

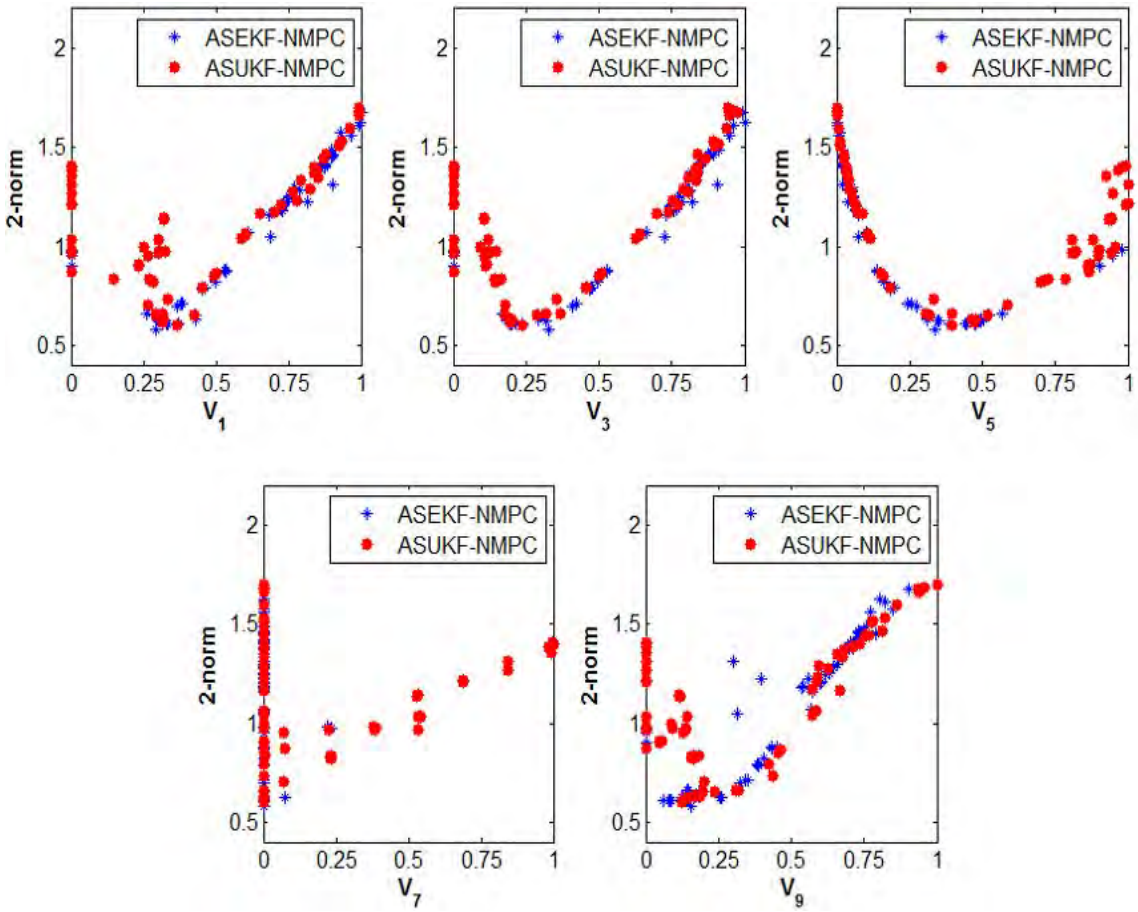


Figure 8.1: *Level Diagrams* of ASEKF-NMPC and ASUKF-NMPC controllers at the transient response.

The pendulum was allowed time to minimize the possible steady-state errors in the 2% band of the final value. Later on, the output step disturbance was introduced to perturb the pendulum from the upright position. From the plot of V_4 in Figure 8.2, ASEKF-NMPC

control efforts seems to be more aggressive to counter the effect of the disturbance than those of ASUKF-NMPC. This leads to ASEKF-NMPC achieving better tracking performance after the occurrence of the disturbance. See V_2. Therefore, ASEKF-NMPC achieves good disturbance rejection performance compared to ASUKF-NMPC, but at the cost of large control efforts. On the other hand, ASUKF-NMPC tries to use minimum control efforts to recover from the disturbance. Therefore, the control efforts are not strong enough to fully stabilize the pendulum in the upright position. In this case also, ASUKF seems to provide the NMPC with relatively good state estimates compared to ASEKF. This is apparent from V_{10} since ASEKF-NMPC has some points close to the upper bound than ASUKF-NMPC. From V_6 and V_8, both controllers seem to have almost similar system's response and peak overshoot performances. In general, the ASEKF-NMPC controller gives a better disturbance rejection performance.

However, ASUKF-NMPC seems to have a point with the 2-norm lower than any other point of ASEKF-NMPC. Therefore, it is logical that the designer may choose to implement ASUKF estimator in state estimation-based NMPC to achieve good disturbance rejection performance.

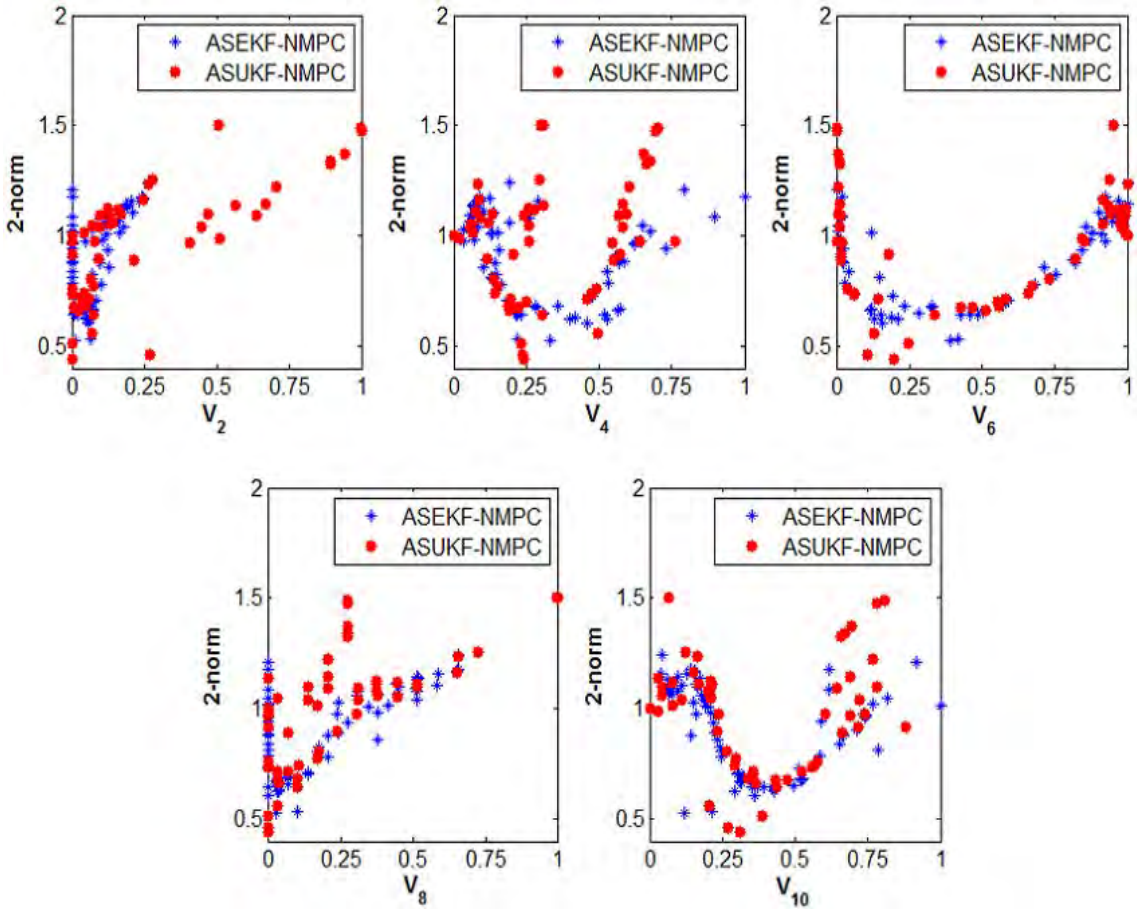


Figure 8.2: *Level Diagrams* of ASEKF-NMPC and ASUKF-NMPC at the steady-state response.

Figures 8.3 and 8.4 show the plots of the ASEKF-NMPC and ASUKF-NMPC tuning parameters, respectively. From all the plots of both figures, the points seem to be distributed over the entire range of values with no range which is better than any other. This confirms

the fact that in general, it is difficult to tune the NMPC control schemes [27, 34].

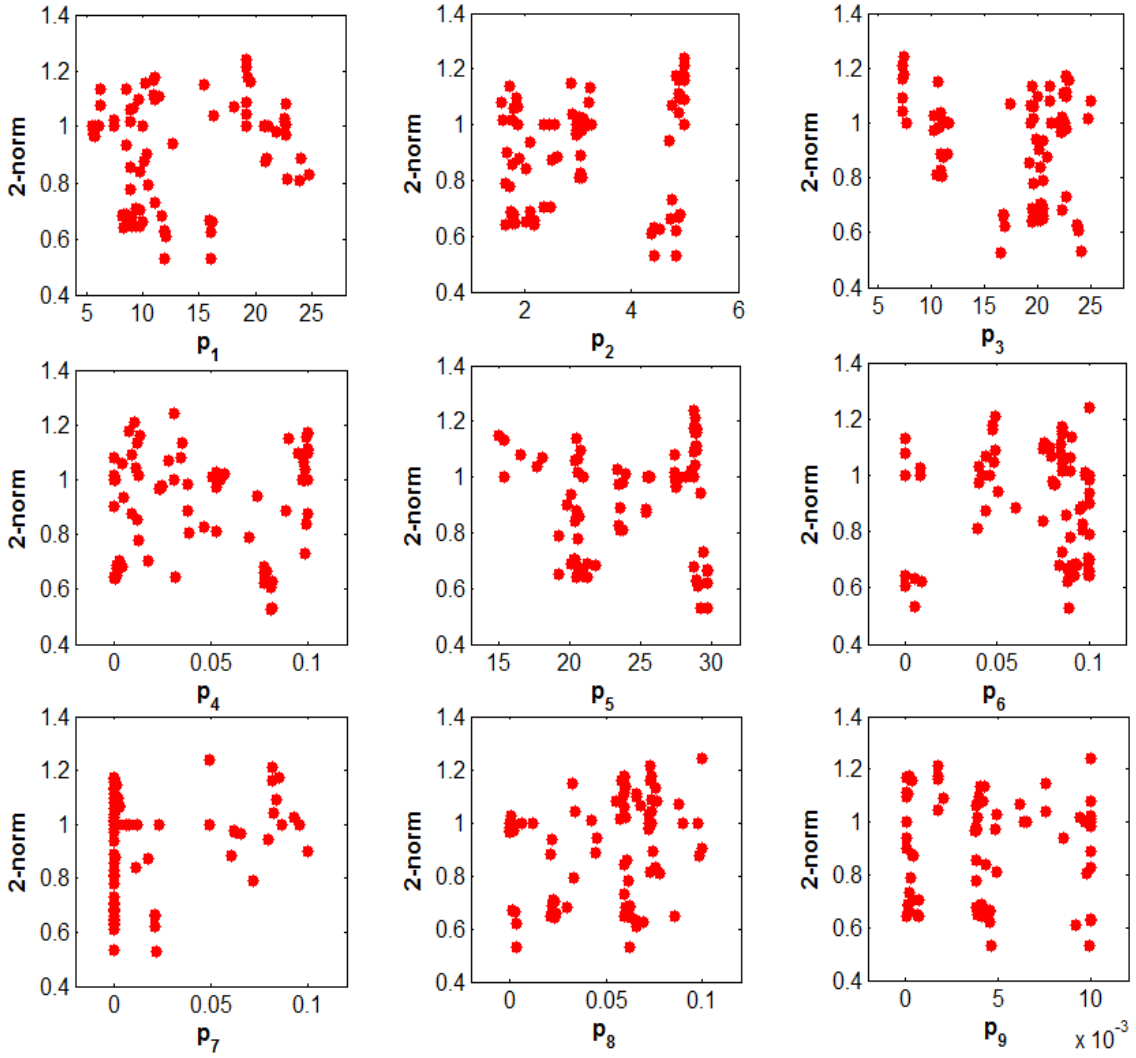


Figure 8.3: *Level Diagrams* of ASEKF-NMPC tuning parameters. p_1, p_2 and p_3 are scalars multiplying elements of $\mathbf{Q}$, p_4 is scalar multiplying element of $\mathbf{R}$, p_5 is N_p. p_6, p_7 and p_8 are scalars multiplying elements of $\mathbf{Q_a}$ and p_9 is scalar multiplying elements of $\mathbf{R_a}$.

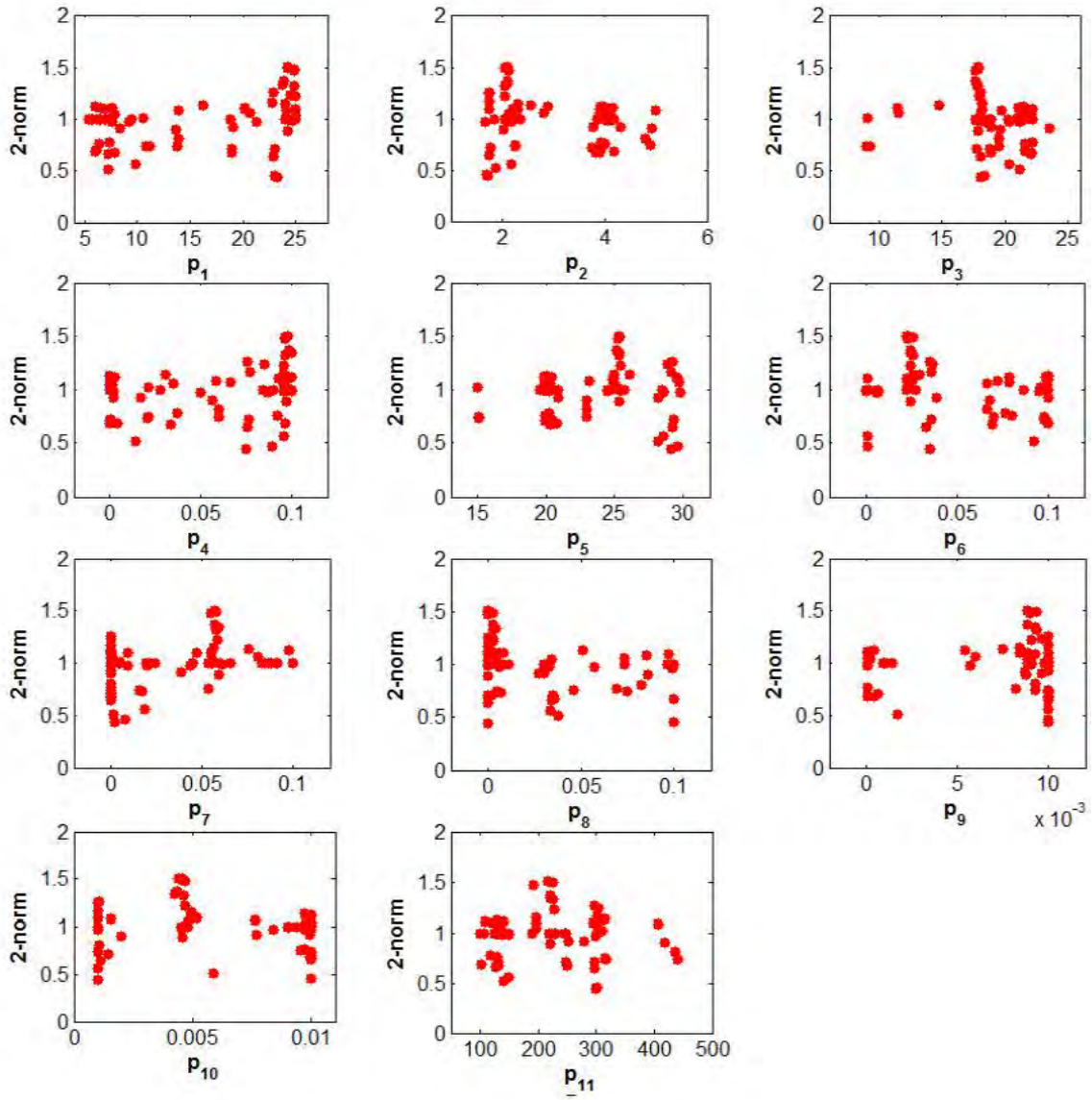


Figure 8.4: *Level Diagrams* of ASUKF-NMPC tuning parameters. p_1, p_2 and p_3 are scalars multiplying elements of $\mathbf{Q}$, p_4 is scalar multiplying element of $\mathbf{R}$, p_5 is N_p. p_6, p_7 and p_8 are scalars multiplying elements of $\mathbf{Q_a}$ and p_9 is scalar multiplying elements of $\mathbf{R_a}$. p_{10} and p_{11} are α and β, respectively.

8.2.2 System Time Responses

In these subsection, the information from the *Level Diagrams* is used for tuning the ASEKF-NMPC and ASUKF-NMPC controllers. The method of using the information from the *Level Diagrams* to tune the NMPC controllers is called the MOO tuning method. The time responses of the controllers are given to show how the controllers perform in time domain. The time responses are of particular importance to the controller designer since they show important control measures such as overshoots, settling time and oscillations on the output and saturation of the input [27].

The points for ASEKF-NMPC and ASUKF-NMPC controllers that give the minimum 2-norms to all objectives were selected from the *Level Diagrams*. The tuning parameter values and cost objective values that correspond to these points in the *Level Diagrams* are

tabulated as shown in Table 8.1. Given these Pareto information, the time responses of the ASEKF-NMPC and ASUKF-NMPC controllers were generated through simulation runs and they are shown in Figures 8.5 through 8.8.

Controller	Norm Values	Input Parameters	Cost Objectives Values
ASEKF-NMPC	0.5762	$p_1 = 16.01$ $p_2 = 4.846$ $p_3 = 16.50$ $p_4 = 0.08086$ $p_5 = 29$ $p_6 = 0.08949$ $p_7 = 0.02209$ $p_8 = 0.003591$ $p_9 = 0.004613$	$V_1 = 0.2932$ $V_2 = 0.3274$ $V_3 = 0.3376$ $V_4 = 0.0007748$ $V_5 = 0.1578$
ASUKF-NMPC	0.6008	$p_1 = 23.22$ $p_2 = 1.718$ $p_3 = 18.05$ $p_4 = 0.07499$ $p_5 = 29$ $p_6 = 0.03556$ $p_7 = 0.002462$ $p_8 = 0.0001$ $p_9 = 0.009993$ $p_{10} = 0.001$ $p_{11} = 299$	$V_1 = 0.3640$ $V_2 = 0.2389$ $V_3 = 0.3949$ $V_4 = 0$ $V_5 = 0.1242$

Table 8.1: Tuning Parameters and Cost Values for ASEKF-NMPC and ASUKF-NMPC controllers. p_1, p_2 and p_3 are scalars multiplying elements of $\mathbf{Q}$, p_4 is scalar multiplying element of $\mathbf{R}$, p_5 is N_p. p_6, p_7 and p_8 are scalars multiplying elements of $\mathbf{Q_a}$ and p_9 is scalar multiplying elements of $\mathbf{R_a}$. p_{10} and p_{11} are α and β, respectively.

ASEKF-NMPC Time Responses

The time response plots of the ASEKF-NMPC controller are shown in Figure 8.5 and the corresponding control input in Figure 8.6. From all plots of Figure 8.5 it can be seen that the state estimates are very close to the true states of the system. This implies that the ASEKF estimator implemented in the ASEKF-NMPC controller is able to produce good estimation information (i.e., state, disturbance and model mismatch estimates) for the controller. The controller is able to swing the pendulum to the vicinity of the upright position in two swings, and with no apparent overshoots in the face of 5% parametric model mismatch.

At $t = 7sec$ of the simulation time, an output step disturbance was introduced into the system to perturb the pendulum from the upright position. The ASEKF-NMPC controller was able to reject the disturbance and therefore let the system to return to the upright position. Figure 8.6 shows the time response of the control input generated by the ASEKF-NMPC controller for the inverted pendulum system.

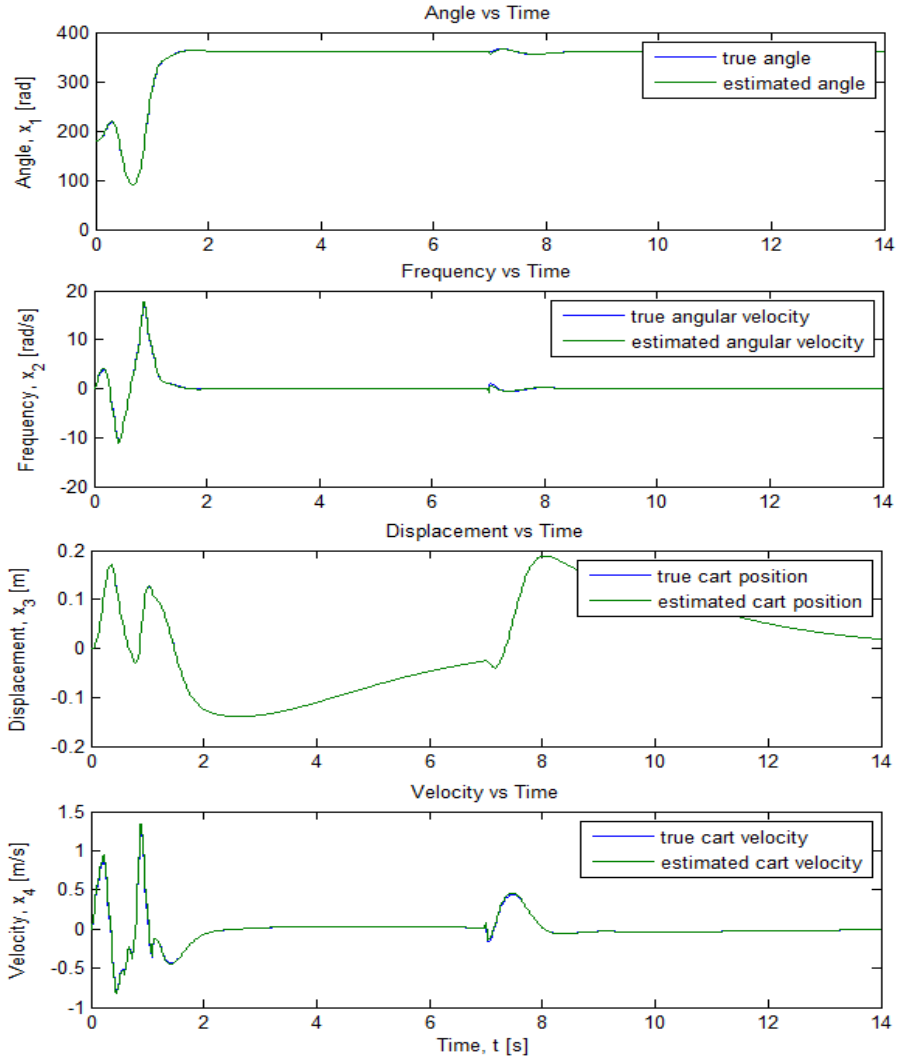


Figure 8.5: Time responses of the true and estimated states of ASEKF-NMPC controller.

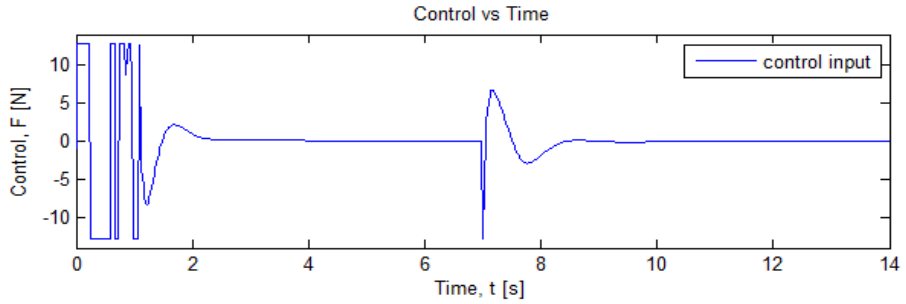


Figure 8.6: The control input force corresponding to time responses in Figure 8.5.

ASUKF-NMPC Time Responses

The time response plots of the ASUKF-NMPC controller are shown in Figures 8.7 and 8.8. Looking at all the plots of Figure 8.7, the ASUKF estimator implemented in the formulation of the ASUKF-NMPC gives accurate state estimates. From Figures 8.5 and 8.7, it can be seen that the ASEKF and ASUKF algorithms are competing in achieving good state estimates. Both of them were able to suppress the effects of 5% parameter plant-model mismatches. Furthermore, both of them are able to reject the output step disturbance.

Looking more closely to the first plot of Figure 8.7, ASUKF-NMPC is able to swing the pendulum to the upright position in three swings. Therefore, implementing ASEKF in NMPC leads to faster response of the system. In contrast, implementing ASUKF in NMPC leads to slower response. In general, both controllers give the desired system response.

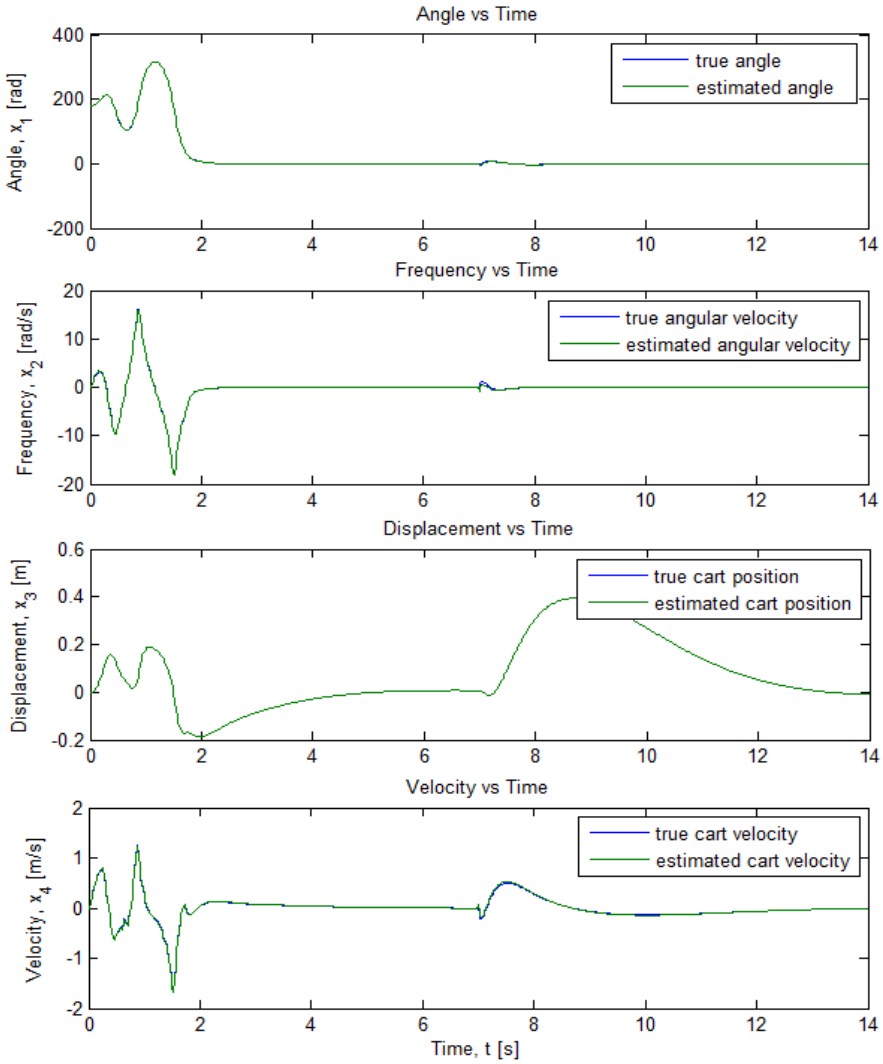


Figure 8.7: True and estimated states generated using ASUKF-NMPC controller.

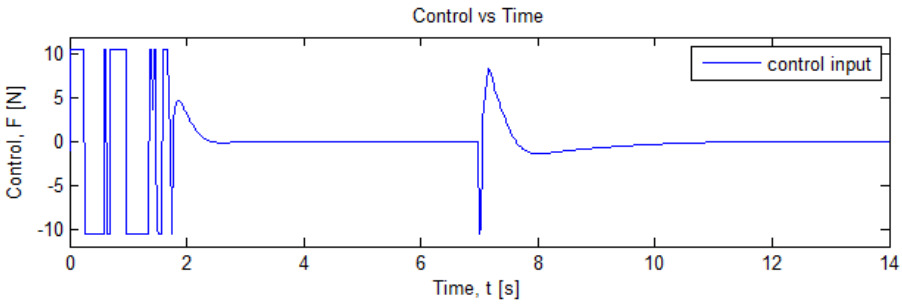


Figure 8.8: The control input force corresponding to time responses in Figure 8.7.

Chapter 9

Conclusions

9.1 Conclusion

The goal of the study was to analyze the closed-loop performances ASEKF and ASUKF estimators when implemented in the NMPC formulation. Each state estimation method was coupled to the same NMPC controller to form the ASEKF-NMPC and ASUKF-NMPC controllers. The resulted controllers were validated by applying them for the feedback control of the magnetic levitation system. The validation results showed that the two controllers were able to provide satisfactory setpoint tracking, disturbance rejection and account for plant-model mismatches. The controllers were further applied for the angular position control of the inverted pendulum mounted on a cart system for comparative analysis. The control system was perturbed with different error sources: output step disturbance and 5% parametric plant-model mismatch. Output step disturbance was introduced to the system to disturb the pendulum from its upright position while the 5% mismatch was applied to the parameters of the model of the controlled system throughout the simulation.

To facilitate fair analysis, Pareto front ranking method was chosen as evaluation method whereby the cost functions were defined according to the author's preferences. The cost functions served as performance markers for analyzing performance of ASEKF and ASUKF in NMPC formulation in multidimensional space. From the *Level Diagrams* of the transient response, ASUKF estimator gave relatively better state estimation accuracy which let better control input and setpoint tracking performances of the ASUKF-NMPC. In contrast, ASEKF-NMPC seemed to have a faster response and less overshoots. Furthermore, ASEKF-NMPC has a point with the lowest 2-norm to all objectives. That made implementing ASEKF for the design of state estimation-based NMPC more preferable than implementing ASUKF. From the *Level Diagrams* of the steady-state response, ASEKF-NMPC controller seemed to achieve better disturbance rejection performance compared to ASUKF-NMPC, at larger control efforts. In contrast, ASUKF-NMPC has more points with the lowest 2-norm for all the objectives.

From this evaluation study, a number of conclusions can be drawn. Firstly, the ASUKF algorithm is well suited for designing state estimation-based NMPC for inverted pendulum systems and many other related systems, in which good setpoint tracking at the cost of

small control efforts is desired. Secondly, if the faster response with less peak overshoots is desired, then ASEKF estimator may be a better choice.

9.1.1 Future Work

This work could be extended to the Pareto analysis of the closed-loop performances of more than two state estimation methods in the formulation of state estimation based NMPC controller. For instance, EKF, UKF, Ensemble Kalman Filter (EnKF), Extended Luenberger Observer (ELO) and Moving Horizon Estimator (MHE) can be analyzed. This multiobjective evaluation will provide the controller designer with a wider range of state estimation methods from which to choose the best for the design. The issue of computational burden of each method has to be taken into consideration as one of the objectives since it is very important for practical systems. Furthermore, the population size of the NSGA-II optimizer has to be increased for the better visualization of the Pareto fronts (i.e., greater than 250).

Bibliography

[1] R. Van Der Merwe, "Sigma-point kalman filters for probabilistic inference in dynamic state-space models," Ph.D. dissertation, Oregon Health & Science University, 2004.

[2] P. J. Fleming and R. Purshouse, "Genetic algorithms in control systems engineering," *Research Report - University of Sheffield Department of Automatic Control and Systems Engineering*, 2001.

[3] T. Koetje, "Multi-objective model predictive control of multivariable systems," Master's thesis, University of Cape Town, 2010.

[4] A. Alaniz, "Model predictive control with application to real-time hardware and guided parafoil," Master's thesis, Massachusetts Institute of Technology, 2004.

[5] L. Grune and J. Pannek, *Nonlinear Model Predictive Control: Theory and Algorithms*. Springer, 2011.

[6] A. Flores-Tlacuahuac, P. Morales, and M. Rivera-Toledo, "Multiobjective nonlinear model predictive control of a class of chemical reactors," *Industrial & Engineering Chemistry Research*, vol. 51, no. 17, pp. 5891–5899, 2012.

[7] A. Acikmese and J. M. Carson III, "A nonlinear model predictive control algorithm with proven robustness and resolvability," in *American Control Conference, 2006*. IEEE, 2006, pp. 7–pp.

[8] N. C. Jacob and R. Dhib, "Unscented kalman filter based nonlinear model predictive control of a ldpe autoclave reactor," *Journal of Process Control*, vol. 21, no. 9, pp. 1332–1344, 2011.

[9] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control engineering practice*, vol. 11, no. 7, pp. 733–764, 2003.

[10] J. S. Qin and T. A. Badgwell, "An overview of nonlinear model predictive control applications," in *Nonlinear model predictive control*. Springer, 2000, pp. 369–392.

[11] T. Mohohlo, "Lecture notes: Eee4093 - model predictive control (mpc)," 2011.

[12] M. J. Tenny, J. B. Rawlings, and S. J. Wright, "Closed-loop behavior of nonlinear model predictive control," *AIChE Journal*, vol. 50, no. 9, pp. 2142–2154, 2004.

[13] G. Marafioti, S. Olaru, and M. Hovd, "State estimation in nonlinear model predictive control, unscented kalman filter advantages," in *Nonlinear Model Predictive Control*. Springer, 2009, pp. 305–313.

[14] M. Diehl, H. J. Ferreau, and N. Haverbeke, "Efficient numerical methods for nonlinear mpc and moving horizon estimation," in *Nonlinear model predictive control*. Springer, 2009, pp. 391–417.

[15] B. W. Bequette, "Non-linear model predictive control: A personal retrospective," *The Canadian Journal of Chemical Engineering*, vol. 85, no. 4, pp. 408–415, 2007.

[16] S. Papp, K. György, A. Kelemen, and L. Jakab-Farkas, "Applying the extended and unscented kalman filters for nonlinear state estimation," in *Inter-Eng 2012 Conference Proceedings, Tg Mureş*, 2012, pp. 233–239.

[17] U. Maeder, "Augmented models in estimation and control," Ph.D. dissertation, ETH Zurich University, 2010.

[18] S. S. Haykin, *Kalman filtering and neural networks*. Wiley Online Library, 2001.

[19] R. Kandepu, B. Foss, and L. Imsland, "Applying the unscented kalman filter for nonlinear state estimation," *Journal of Process Control*, vol. 18, no. 7, pp. 753–768, 2008.

[20] Y. Zhou, J. Xu, Y. Jing, and G. M. Dimirovski, "The unscented kalman filtering in extended noise environments," in *American Control Conference, 2009. ACC'09.* IEEE, 2009, pp. 1865–1870.

[21] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[22] I. M. Moreno, "A comparative performance evaluation of nonlinear observers for a fed-batch evaporative crystallization process," Master's thesis, Delft University of Technology, 2011.

[23] J. Prakash and R. Senthil, "Design of observer based nonlinear model predictive controller for a continuous stirred tank reactor," *Journal of Process Control*, vol. 18, no. 5, pp. 504–514, 2008.

[24] R. Huang, S. C. Patwardhan, and L. T. Biegler, "Robust extended kalman filter based nonlinear model predictive control formulation," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on.* IEEE, 2009, pp. 8046–8051.

[25] M. Grewal and A. Andrews, "Applications of kalman filtering in aerospace 1960 to the present [historical perspectives]," *Control Systems, IEEE*, vol. 30, no. 3, pp. 69–78, June 2010.

[26] M. Rhuby and Y. Gu, "Interactive robotics letters: Understanding nonlinear kalman filters, part ii: An implementation guide," 2013.

[27] Y. Wu, D. Hu, M. Wu, and X. Hu, "Unscented kalman filtering for additive noise case: augmented vs. non-augmented," in *American Control Conference, 2005. Proceedings of the 2005.* IEEE, 2005, pp. 4051–4055.

[28] I. Ahmad and M. A. Javaid, "Nonlinear model & controller design for magnetic levitation system," in *WSEAS International Conference on Signal Processing, Robotics and Automation, 2010. Proceedings of 2010.* WSEAS, 2010, pp. 324–328.

[29] D. Moore, "Optimal controller comparison using pareto fronts," Master's thesis, University of Cape Town, 2010.

[30] C. A. Coello Coello, "Evolutionary multi-objective optimization: a historical view of the field," *Computational Intelligence Magazine, IEEE*, vol. 1, no. 1, pp. 28–36, 2006.

[31] A. Seshadri, "Nsga-ii: A multi-objective optimization algorithm," March 2009, a Documentation File for Matlab Code Implementation. Accessed on 20-Feb-2015. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/10429-nsga-ii--a-multi-objective-optimization-algorithm/content/NSGA-II.zip

[32] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii," *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 2, pp. 284–302, 2009.

[33] E. Zio and R. Bazzo, "Level diagrams analysis of pareto front for multiobjective system redundancy allocation," *Reliability Engineering & System Safety*, vol. 96, no. 5, pp. 569–580, 2011.

[34] X. Blasco, J. Herrero, J. Sanchis, and M. Martínez, "A new graphical visualization of n-dimensional pareto front for decision-making in multiobjective optimization," *Information Sciences*, vol. 178, no. 20, pp. 3908–3924, 2008.

[35] T. Maeba, M. Deng, A. Yanou, and T. Henmi, "Swing-up controller design for inverted pendulum by using energy control method based on lyapunov function," in *Modelling, Identification and Control (ICMIC), The 2010 International Conference on.* IEEE, 2010, pp. 768–773.

Appendix A

ASEKF and ASUKF Estimators Codes

The Matlab codes for the ASEKF and ASUKF estimators are designed specifically for the inverted pendulum on a moving cart. Implementing these codes on another system is easy since the user only has to replace the model of the inverted pendulum with the model of his/her system. Apart from that, nothing else needs to be edited. Furthermore, the Matlab codes for ASEKF and ASUKF estimators for magnetic levitation system and Van der Pol Oscillator system are available within the CD attached to this document.

For further information about the source codes used here, please contact the author directly through his email account, i.e., moeti.sekhonyana@gmail.com.

Appendix B

Paper Submission

Some of the work in this thesis were published on a paper:

- S. Moeti, T. Mohohlo, "State Estimation for Nonlinear Model Predictive Control: EKF and UKF Approaches", **CISSE**, USA, December 2014 (In Press)

Appendix C

Resources Used

All the resourses used are included in a compact disk (CD) that is submitted together with this thesis.

The main files are *VanderPol.m*, *MagLevControl.m*, *InvertedPendulumControl.m*, *nsga_2.m* and *LevelDiagramsTests.m*. *VanderPol.m* was used to validate the proposed ASEKF and ASUKF estimators while *MagLevControl.m* was used to validate the closed performances of ASEKF-NMPC and ASUKF-NMPC controllers. *InvertedPendulumControl.m*, *nsga_2.m* and *LevelDiagramsTests.m* were used to generate the Pareto fronts and *Level Diagrams* results of the ASEKF-NMPC and ASKF-NMPC controllers. Other files including those for the ASEKF, ASUKF, NMPC and others are also included.